



THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)

Présentée et soutenue par :

MARTIN Arnaud

Le 04 Octobre 2012 à 10H30

Titre :

Évolution de profils Multi-Attributs, par Apprentissage Automatique et Adaptatif dans un système de Recommandation pour l'Aide à la Décision

École doctorale et discipline ou spécialité :

ED MITT : Domaine STIC : Intelligence Artificielle

Unité de recherche :

Institut de Recherche en Informatique de Toulouse (IRIT)

Directeur(s) de Thèse :

Professeur
Maître de Conférences

Université Toulouse Capitole
Université Toulouse 3

ZARATE Pascale
CAMILLERI Guy

Rapporteurs :

Professeur
Maître de Conférences (HDR)

École Centrale Paris
Université Lille 1

MOUSSEAU Vincent
ROUILLARD José

Autre(s) membre(s) du jury :

Président du Jury Professeur
Directeur Industriel

Université Toulouse 3
Société PRESTALIA

BOUGHANEM Mohand
REVELAT Philippe

REMERCIEMENTS

Il est malheureusement impossible de remercier tous les anciens collègues et amis qui sont partis et ceux qui sont arrivés pour faire des thèses, des stages, etc. Je tiens malgré tout à remercier quelques personnes en particulier.

ZARATE Pascale, Professeur à l'Université Toulouse I, et membre de l'IRIT (Institut de Recherche en Informatique de Toulouse). Qu'elle trouve ici tous mes remerciements pour m'avoir offert l'opportunité de réaliser ce doctorat à travers cette thèse CIFRE. Je la remercie de m'avoir soutenu, de m'avoir éclairé quand j'étais perdu, et de m'avoir supporté, quand j'ai rendu mon travail avec un délai supplémentaire ou quand j'oubliais un de nos rendez-vous. Je la remercie de tous ses conseils avisés, aussi bien sur la thèse que sur la société et son fonctionnement. Je la remercie de sa présence continue et d'avoir si bien organisé cette thèse. Je n'aurais pu rêver meilleure directrice de thèse.

CAMILLERI Guy, Maître de Conférence à l'Université Toulouse III (Université Paul Sabatier), qui a été mon co-encadrant pour cette thèse CIFRE, est grandement remercié pour les nombreuses discussions qui ont été importantes pour la réussite de cette thèse. Discussions sur la thèse mais très souvent, discussions philosophiques sur la vie et le monde. Je le remercie de m'avoir fait partager son expérience, sa bonne humeur permanente, de m'avoir soutenu, encouragé, et d'avoir été un pilier et une source de motivation continue pour moi tout au long de ces 3 années.

REVELAT Philippe, directeur de l'entreprise Prestalia, qui m'a offert la possibilité de réaliser cette thèse CIFRE au sein de son entreprise. Qu'il trouve ici mes remerciements pour sa patience, sa compréhension et ses conseils.

GRISTI Benjamin, directeur technique de l'entreprise Prestalia, je le remercie de la place qu'il m'a faite dans son équipe, de son soutien, de ses expériences, et de sa bonne humeur.

MOUSSEAU Vincent, Professeur à l'École Centrale Paris, pour avoir accepté de faire partie de mon Jury.

ROUILLARD José, Maître de Conférence (HDR), pour avoir accepté de faire partie de mon Jury.

BOUGHANEM Mohand, Professeur à l'Université Toulouse III (Université Paul Sabatier), pour avoir accepté de faire partie de mon Jury, et de le présider.

Mes plus vifs remerciements vont également aux membres de l'équipe IC3 qui m'ont accueilli, et aux membres de l'équipe SMAC, qui par la suite m'ont également chaleureusement accueilli.

Enfin, un grand merci à tous les « supporters » qui ont contribué à maintenir mon moral au plus haut. Il y a bien sûr ma compagne Melissa que j'ai rencontré pendant ce doctorat à cette place pour partager un chocolat et qui m'a encouragé tout particulièrement pour la rédaction de ce manuscrit où de nombreuses fois nous dûmes rester chez moi car je devais travailler. Merci à elle pour m'avoir soulagé sur cette partie, la plus dure, qui a été allégée par sa bonne humeur et sa compréhension. Mes amis : David, pour tous ces moments de joie partagée et de parties de jeux-vidéos, et également pour sa relecture de cette thèse ; Sylvain pour les soirées à discuter et à parler de la vie et des filles ; Angélique pour m'avoir offert un soutien lors de mes premiers jours dans l'entreprise, et par la suite, pour les bons moments passé ; Guylaine pour être restée une amie toujours présente depuis toutes ses années, merci pour son support et ses conseils quand j'en avais besoin ; Julien, qui bien que parti loin d'ici, est resté un très bon ami.

Et pour finir, je remercie ma famille, qui a dû me supporter toutes ses années, et pas seulement pendant ce doctorat. Je les remercie pour leur amour et leur présence continuelle.

RESUMÉ

La prise en compte des profils utilisateurs ainsi que leurs évolutions, dans le domaine de l'aide à la décision, constitue actuellement dans la communauté des SIAD (Systèmes Interactifs d'Aide à la Décision) un enjeu important. En effet, la prise en compte du contexte lors de la décision est actuellement émergente pour les SIAD. Ces systèmes d'assistance offrent ainsi des conseils aux utilisateurs en se basant sur leur profil, qui représente leurs préférences à travers une liste de critères valués. Les principales contraintes viennent du fait qu'il est nécessaire que le système puisse amener de l'information pertinente de manière continue. Cela oblige donc à faire évoluer les profils des utilisateurs en fonction de leurs actions. Pour cela, le système ne doit pas seulement « comprendre » ce que l'utilisateur aime, mais également pourquoi. De plus, l'aide apportée aux utilisateurs évoluera donc dans le temps et également par rapport à l'utilisateur. Ainsi l'utilisateur aura à sa disposition une sorte d'assistant personnalisé.

L'objectif du travail consiste à apporter une aide à l'activité de l'utilisateur en fonction de son profil. Pour cela, nous proposons de mettre en œuvre et de développer des algorithmes, basés sur des techniques issues du domaine de l'apprentissage, afin de faire évoluer le profil d'un utilisateur en fonction de ses actions. L'aide apportée à l'utilisateur par le système évoluera aussi en fonction de l'évolution de son profil. Le problème à traiter pour l'utilisateur est un problème de prise de décision. Pour ce problème, une assistance est apportée à l'utilisateur, et celle-ci se fait par un affinage des solutions potentielles. Cet affinage est effectué grâce à la mise en place d'un tri (ranking) évolutif des solutions qui sont présentées à l'utilisateur en fonction de son/ses profils.

La réalisation d'un tel système nécessite l'articulation des trois principaux domaines de recherche ; qui sont l'Aide à la Décision multicritère, la Décomposition et Agrégation de préférence, et l'Apprentissage automatique. Les domaines de l'Aide à la Décision multicritère et de la Décomposition et Agrégation de préférence peuvent être aussi rassemblés en tant que Procédure d'Agrégation Multicritère (PAMC). Certaines méthodes d'Aide à la Décision multicritère sont mises en place ici et utilisent les données du profil afin d'apporter la meilleure aide possible à l'utilisateur. La décomposition est utilisée pour caractériser un objet afin de fournir à l'apprentissage les données nécessaires à son fonctionnement. L'agrégation quant à elle sert à obtenir une note sur un objet, et cela selon le profil de l'utilisateur, afin de pouvoir effectuer un classement (ranking). L'apprentissage sert à faire évoluer les profils des utilisateurs afin d'avoir toujours un profil représentant le plus fidèlement possible les préférences des utilisateurs. En effet les préférences des utilisateurs

évoluant dans le temps, il est nécessaire de traiter ces changements afin d'adapter les réponses à apporter à l'utilisateur.

Les contributions de cette thèse portent tout d'abord sur la définition, la construction et l'évolution d'un profil utilisateur (profiling évolutif) en fonction des actions explicites et implicites de l'utilisateur. Ce profiling évolutif est mis en œuvre au sein d'un système de recommandation utilisable sans base d'apprentissage, de manière synchrone et totalement incrémentale, et qui permet aux utilisateurs de changer rapidement de préférences et même d'être incohérents (rationalité limitée). Ce système, qui vient en complément d'un système de Recherche Information, a pour objectif d'établir un ordre total sur une liste d'éléments proposés à l'utilisateur (ranking), et ce en concordance avec les préférences de l'utilisateur. Ces contributions consistent également à la définition de techniques qui permettent d'apporter des parties de solutions à des verrous technologiques comme la désagrégation de critères et la prise en compte d'un nombre variable de critères dans le processus d'aide à la décision interactif, et ce sans définir au préalable de famille cohérente de critères sur laquelle est basée la décision. Plusieurs cadres applicatifs ont été définis afin d'évaluer le système par rapport à d'autres systèmes, mais également afin de tester ses performances de manière hors ligne avec des vraies données utilisateurs, ainsi qu'en ligne, en utilisant directement le système.

MOTS-CLÉS : Systèmes de recommandation, Analyse MultiAttribut, Systèmes d'aide à la Décision, Désagrégation, Agrégation, Apprentissage, Auto-adaptation.

ABSTRACT

Considering user profiles and their evolutions, for decision support is currently in the community of DSS (Decision Support Systems) an important issue. Indeed, the inclusion of context in the decision is currently emerging for DSS. Indeed the system offers advice to users based on their profile, which represents their preferences through a list of valued criteria. The main constraints come from the fact that the system need to continuously bring relevant information. It therefore requires changing user profiles thanks to their actions. So, the system must not only "understand" what the user likes, but also why. The users' assistance will evolve over time and therefore with the user. Thus the user has at his disposal a kind of personal assistant.

The objective of this work is to provide assistance to the user's activity according to his profile. The objective is to develop an algorithm based on automatic techniques, in order to change the profile of a user based on his actions. The assistance provided to the user by the system will evolves according to the evolution of its profile. The problem addressed to the user is a problem of decision making. For this problem, assistance is provided to the user, and it is a refinement of potential solutions. This refining is done through the establishment of scalable scheduling solutions that are presented to the user depending on his / her profile.

The realization of such a system requires the articulation of the three main areas of research which are the Multi-Criteria Decision Support, the Disaggregation and Aggregation of preferences, and Machine Learning. The fields of Decision Support and Multi Disaggregation and Aggregation preference can also be assembled as Multi-Criteria Aggregation Process (PAMC). Some methods of Multicriteria Decision Support are set up here and use profile data to provide the best possible support to the user. The decomposition is used to characterize an object to provide data to the learning algorithm required for its operation. Aggregation serves to score an object according to the user profile in order to rank the selected items. Machine Learning is used to change user profiles in order to always have a profile representing as closely as possible the preferences of users. Indeed user preferences change over the time, it is necessary to address these changes in order to adapt the answers to the user.

The contributions of this thesis are firstly, the definition, construction and evolution of a user profile (evolutionary profiling) based on explicit and implicit user's actions. This evolutionary profiling is implemented within a recommender system usable without learning base, synchronously and completely incremental, and that allows users to quickly change their preferences and even to be inconsistent (bounded rationality). This system,

which complements an Information System Research, aims to establish a total order on a list of items proposed to the user (ranking) and in accordance with his preferences. These also include the definition of techniques used to make parts of solutions to technological challenges as the disintegration of criteria and the inclusion of a variable number of criteria in the process of interactive decision support, and this without firstly defining coherent family of criteria on which the decision is based. Several application frameworks have been developed to evaluate the system and compare it to other systems, but also to test its performance with real user data in an offline mode, and in an online mode using directly the system.

KEYWORDS : Recommender Systems, Multi-Attribute Analysis, Decision Support Systems, Disaggregation, Aggregation, Learning Systems, Auto-adaptation.

Table des Matières

REMERCIEMENTS	2
RESUMÉ	4
ABSTRACT	6
TABLE DES MATIERES	8
TABLE DES FIGURES	12
TABLE DES TABLEAUX	12
TABLE DES GRAPHIQUES	13
TABLE DES FORMULES.....	14
INTRODUCTION	16
PARTIE I. ÉTAT DE L'ART	20
CHAPITRE I. SYSTEMES DE RECOMMANDATION	20
I.1. PROFILS UTILISATEURS.....	20
I.1.A. Définition d'un profil	21
I.1.B. Profils pour assistants personnels.....	22
I.2. FILTRAGE D'INFORMATIONS.....	25
I.2.A. Architecture générale d'un système de filtrage d'informations.....	25
I.2.B. Fonction de décision	26
I.2.C. Le filtrage adaptatif incrémental	27
I.3. SYSTEMES DE RECOMMANDATION.....	28
I.3.A. Recommandation collaborative (collaborative filtering).....	29
I.3.B. Recommandation sur le contenu (content based).....	30
I.3.C. Recommandation hybride.....	32
I.3.D. Autres systèmes de recommandations	33
I.4. CONCLUSION	33
CHAPITRE II. AIDE A LA DECISION MULTICRITERE / MULTIATTRIBUT	36
II.1. AIDE A LA DECISION MULTICRITERE	36
II.1.A. Les critères.....	38
II.1.A.i. Vrai critère	38
II.1.A.ii. Quasi-critère	39
II.1.A.iii. Pseudo-critères.....	40
II.1.A.iv. Incomparabilité.....	40
II.1.B. Éléments taxinomiques.....	41
II.1.C. Méthode Interactive d'Aide à la Décision.....	44

II.2.	THEORIE MULTI-ATTRIBUTS.....	46
II.3.	CONCLUSION	48
CHAPITRE III. PRISE EN COMPTE ET EVOLUTION DE PROFILS/PREFERENCES.....		49
III.1.	APPRENTISSAGE DE PROFILS/PREFERENCES	49
III.1.A.	<i>Exploitation des profils/préférences.....</i>	<i>50</i>
III.1.A.i.	La classification.....	50
III.1.A.ii.	Le ranking	50
III.1.B.	<i>Les réseaux neuronaux</i>	<i>51</i>
III.1.C.	<i>Apprentissage par renforcement pour les profils utilisateurs</i>	<i>54</i>
III.2.	DESAGREGATION ET AGREGATION AVEC INDEPENDANCE DES CRITERES.....	56
III.2.A.	<i>Opérateurs de désagrégation.....</i>	<i>56</i>
III.2.A.i.	La décomposition additive.....	57
III.2.A.ii.	La décomposition multiplicative.....	57
III.2.A.iii.	La décomposition multilinéaire	57
III.2.A.iv.	La structure décomposable	58
III.2.A.v.	La décomposition additive non transitive	58
III.2.B.	<i>Opérateurs d'agrégation.....</i>	<i>59</i>
III.2.B.i.	La somme pondérée ou moyenne arithmétique pondérée.....	59
III.2.B.ii.	Le minimum et maximum pondérés.....	60
III.2.B.iii.	La somme pondérée ordonnée (OWA)	61
III.2.C.	<i>Comparaison d'éléments.....</i>	<i>61</i>
III.3.	DESAGREGATION/AGREGATION AVEC INTERACTION ENTRE LES CRITERES.....	63
III.3.A.	<i>Méthode par système expert / règles de production</i>	<i>64</i>
III.3.B.	<i>Mesures floues</i>	<i>64</i>
III.3.C.	<i>Intégrales de Choquet</i>	<i>66</i>
III.4.	CONCLUSION	75
CONCLUSION GENERALE DE L'ETAT DE L'ART		77
PARTIE II. CONTRIBUTIONS		79
CHAPITRE IV. MODELE DE PRISE EN COMPTE DE PROFIL (PROFILING)		79
IV.1.	CONTEXTE ET CONTRAINTES.....	79
IV.2.	MODELE PROPOSE.....	81
IV.2.A.	<i>Système de Recommandation.....</i>	<i>82</i>
IV.2.B.	<i>Profils utilisateurs</i>	<i>84</i>
IV.2.C.	<i>Filtrage d'information</i>	<i>86</i>
IV.2.D.	<i>Apprentissage des préférences.....</i>	<i>87</i>
IV.2.E.	<i>Algorithme général</i>	<i>89</i>
IV.3.	CONCLUSION	93
CHAPITRE V. METHODES DE CALCUL		94
V.1.	INITIALISATION DU SYSTEME	94
V.1.A.	<i>Initialisation sans informations de l'utilisateur</i>	<i>94</i>

V.1.B. Initialisation avec des informations de l'utilisateur.....	95
V.2. ORDONNER LES ELEMENTS	96
V.3. CALCUL DU SCORE	97
V.4. ALGORITHME N°1 : DESAGREGATION A PARTIR D'UNE NOTE	99
V.5. ALGORITHME N°2 : DESAGREGATION A PARTIR DE PAIRES D'OBJETS (PAIRWISE)	100
V.6. CONCLUSION	105
CHAPITRE VI. UTILISATION DES DEPENDANCES/INTERACTIONS ENTRE LES CRITERES	106
VI.1. MISE EN ŒUVRE	107
VI.1.A. Déterminer les contraintes.....	108
VI.1.B. Calcul des utilités	109
VI.1.C. Utilisation des données à l'issue des intégrales de Choquet	110
VI.1.C.i. Méthode directe	111
VI.1.C.ii. Méthode d'utilisation en tant que facteur multiplicateur	112
VI.2. CONCLUSION	113
CONCLUSION GENERALE DES CONTRIBUTIONS.....	115
PARTIE III. EXPERIMENTATIONS.....	118
CHAPITRE VII. METHODOLOGIE.....	118
VII.1. OUTILS DISPONIBLES.....	118
VII.1.A. Mesures de précision de la prédiction de note.....	119
VII.1.B. Mesure de prévision d'utilisation	120
VII.1.C. Mesure d'ordre (ranking mesure)	122
VII.1.C.i. NDPM.....	123
VII.1.C.ii. Mesures de corrélation	125
VII.1.D. Méthode de K-cross fold.....	127
VII.2. SELECTION DES OUTILS D'EVALUATION	128
VII.3. CONCLUSION	128
CHAPITRE VIII. EXPERIMENTATIONS	129
VIII.1. COMPARAISON AVEC AMAZON.....	129
VIII.1.A. Description	129
VIII.1.B. Méthodologie.....	130
VIII.1.C. Résultats	131
VIII.1.D. Conclusion	133
VIII.2. MOVIELENS (BASE DE DONNEES DE TEST)	134
VIII.2.A. Comparaison entre les algorithmes 1 et 2.....	134
VIII.2.A.i. Description	135
VIII.2.A.ii. Méthodologie	135
VIII.2.A.iii. Résultats	137
VIII.2.A.iv. Conclusion.....	138
VIII.2.B. Évaluation de la prise en compte de nombreux critères	139

VIII.2.B.i.	Données	139
VIII.2.B.ii.	Méthodologie.....	139
VIII.2.B.iii.	Résultats	139
VIII.2.B.iv.	Conclusion	142
VIII.3.	MOBIFIZ (PLATEFORME REELLE)	142
VIII.3.A.	Description	142
VIII.3.B.	Méthodologie.....	143
VIII.3.C.	Résultats	145
VIII.3.D.	Conclusion	146
VIII.4.	COMPARAISON AVEC UN SYSTEME MULTI-AGENTS REALISANT DU PROFILING	
UTILISATEUR	146	
VIII.4.A.	Description	147
VIII.4.B.	Méthodologie.....	147
VIII.4.C.	Résultats	148
VIII.4.D.	Conclusion	152
VIII.5.	COMPARAISON AVEC UN SYSTEME EMPLOYANT LA RECOMMANDATION COLLABORATIVE	
	153	
VIII.5.A.	Description	153
VIII.5.B.	Méthodologie.....	154
VIII.5.C.	Résultats	155
VIII.5.D.	Conclusion	156
VIII.6.	COMPARAISON AVEC UN SYSTEME DE RECOMMANDATION HYBRIDE.....	157
VIII.6.A.	Description	157
VIII.6.B.	Méthodologie.....	157
VIII.6.C.	Résultats	158
VIII.6.D.	Conclusion	159
CONCLUSION GENERALE DES EXPERIMENTATIONS		161
CONCLUSION GENERALE DE LA THESE		164
PARTIE IV. ANNEXES		168
MOBIFIZ (MOBIFIZ.FR)		168
Captures d'écran		168
Tableaux des résultats obtenus lors de l'expérimentation		170
MOVIELENS		175
Méthode Rate sur la BDD restreinte.....		175
Méthode Pairwise sur la BDD restreinte		179
Méthode Pairwise sur la BDD riche		184
SYSTEME MULTI-AGENTS		189
Résultats sur BDD restreinte.....		189
Résultats sur BDD riche		194
PARTIE V. REFERENCES BIBLIOGRAPHIQUES		200

Table des Figures

Figure 1. <i>Vrai critère</i>	39
Figure 2. <i>Quasi-critère</i>	39
Figure 3. <i>Pseudo-critère</i>	40
Figure 4. <i>Processus interactif de décision</i>	45
Figure 5. <i>Cycle d'apprentissage de préférences</i>	49
Figure 6. <i>Exemple de réseau de neurones</i>	52
Figure 7. <i>Exemple de nœud</i>	52
Figure 8. <i>Fonctionnement général du système</i>	90
Figure 9. <i>Exemple d'interface en utilisation réelle avec Mobifiz</i>	92
Figure 10. <i>Exemple d'interface en utilisation réelle avec Mobifiz après interaction de l'utilisateur</i>	92
Figure 11. <i>Fonctionnement général du test</i>	130

Table des Tableaux

Tableau 1. <i>Récapitulatif de différents systèmes de recommandation</i>	35
Tableau 2. <i>Taxinomie des problématiques</i>	41
Tableau 3. <i>Matrice de performance</i>	42
Tableau 4. <i>Exemple d'association d'éléments et d'une note</i>	108
Tableau 5. <i>Exemple de représentation sous la forme de Möbius pour 5 critères</i>	110
Tableau 6. <i>Classification des résultats possibles d'une recommandation d'un élément pour l'utilisateur</i>	120
Tableau 7. <i>Exemple de valeur de coefficients de Kendall Tau-b</i>	126
Tableau 8. <i>Moyenne des écarts et coefficient de Kendall Tau-b</i>	131
Tableau 9. <i>Moyenne des écarts et coefficient de Kendall Tau-b en incluant la base de validation dans la base d'apprentissage</i>	132
Tableau 10. <i>Moyenne des écarts et coefficient de Kendall Tau-b, avec des données issues de plusieurs bases d'apprentissage</i>	132
Tableau 11. <i>Moyenne de l'indice de performance en fonction du groupe</i>	145
Tableau 12. <i>Comparaison avec des systèmes de recommandation hybride, en utilisant le coefficient de Spearman Rho</i>	158
Tableau 13. <i>Récapitulatif des tests effectués</i>	162

Table des Graphiques

Graphique 1. Valeur de Kendall Tau-b obtenue par utilisateur par notre système mettant en œuvre l'algorithme Rate	137
Graphique 2. Valeur de Kendall Tau-b obtenue par utilisateur par notre système mettant en œuvre l'algorithme Pairwise	137
Graphique 3. Comparaison de la valeur de Kendall Tau-b obtenue par utilisateur par notre système mettant en œuvre l'algorithme Rate et Pairwise	138
Graphique 4. Valeur de Kendall Tau-b obtenue par utilisateur par notre système mettant en œuvre l'algorithme Pairwise avec la base de données restreinte	140
Graphique 5. Valeur de Kendall Tau-b obtenue par utilisateur par notre système mettant en œuvre l'algorithme Pairwise avec la base de données riche.....	140
Graphique 6. Comparaison de la valeur de Kendall Tau-b obtenue par utilisateur par notre système mettant en œuvre l'algorithme Pairwise, pour une base de données restreinte et une base de données riche	141
Graphique 7. Moyenne de l'indice de performance par utilisateurs.....	145
Graphique 8. Valeur de Kendall Tau-b obtenue par utilisateur par le système multi-agents utilisant une base de données restreinte.....	148
Graphique 9. Valeur de Kendall Tau-b obtenue par utilisateur par notre système mettant en œuvre l'algorithme Rate	149
Graphique 10. Valeur de Kendall Tau-b obtenue par utilisateur par notre système mettant en œuvre l'algorithme Pairwise et utilisant une base de données restreinte.....	149
Graphique 11. Comparaison des valeurs de Kendall Tau-b obtenue par utilisateur pour le système multi-agents et notre système utilisant l'algorithme de Pairwise avec une base de données restreinte.....	150
Graphique 12. Valeur de Kendall Tau-b obtenue par utilisateur par le système multi-agents utilisant une base de données riche	151
Graphique 13. Valeur de Kendall Tau-b obtenue par utilisateur par notre système mettant en œuvre l'algorithme Pairwise et utilisant une base de données riche	151
Graphique 14. Comparaison des valeurs de Kendall Tau-b obtenue par utilisateur pour le système multi-agents et notre système utilisant l'algorithme de Pairwise avec une base de données riche	152
Graphique 15. Valeur du coefficient de corrélation Kendall Tau QCF, QCFA, et CF en fonction de la valeur de δ	155
Graphique 16. Valeur de Kendall Tau obtenue par utilisateurs par notre système mettant en œuvre l'algorithme Pairwise et utilisant une base de données riche	156

Table des formules

(1)	47
(2)	47
(3)	55
(4)	55
(5)	57
(6)	57
(7)	57
(8)	58
(9)	58
(10)	58
(11)	59
(12)	59
(13)	59
(14)	60
(15)	60
(16)	60
(17)	61
(18)	66
(19)	71
(20)	73
(21)	74
(22)	74
(23)	75
(24)	94
(25)	95
(26)	97
(27)	99
(28)	100
(29)	103
(30)	103
(31)	103
(32)	104
(33)	104
(34)	104
(35)	111

(36)	111
(37)	112
(38)	112
(39)	119
(40)	119
(41)	121
(42)	121
(43)	121
(44)	122
(45)	123
(46)	123
(47)	124
(48)	124
(49)	124
(50)	124
(51)	125
(52)	125
(53)	126
(54)	126
(55)	126
(56)	144

Introduction

Le but de cette thèse consiste à définir un système d'aide à l'utilisateur afin de l'assister lors de prises de décision, et cela au travers d'un système de Recommandation. Ce système vient en complément d'un système de Recherche d'Information (RI), qui va effectuer des requêtes correspondant aux demandes de l'utilisateur et le système de recommandation proposé ici, va trier les éléments récupérés par le système de RI. Notre système n'effectue pas de requêtes. En effet, le système que nous proposons a pour objectif d'effectuer des tâches de *profiling*, afin d'apprendre et de faire évoluer les profils des utilisateurs en fonction de leurs actions. Le système va ensuite utiliser ces profils utilisateurs afin de calculer un score représentant une estimation de la pertinence d'un élément pour l'utilisateur. La liste d'éléments à présenter à l'utilisateur possède ainsi un préordre total entre les éléments et ces éléments sont donc triés en fonction de cet ordre (en fonction du score de chaque élément). Ainsi l'utilisateur dispose d'un système ordonnant des éléments en fonction de ses préférences, qui évoluent dans le temps.

Notre problématique est multiple. Elle consiste premièrement à définir et mettre en œuvre un système d'assistance à l'utilisateur qui soit :

- **Évolutif et interactif**, c'est-à-dire que les profils des utilisateurs évolueront dans le temps à travers les actions des utilisateurs, et l'assistance fournie évoluera de la même manière. La prise en compte des profils utilisateurs ainsi que leurs évolutions, constitue actuellement dans la communauté des Systèmes d'Informations pour l'Aide à la Décision (SIAD) un enjeu important [BRE et al. 09] ;
- Mettant en œuvre des principes issus des domaines **multicritère/multiattribut** et cela en faisant varier dynamiquement la quantité et les critères/attributs pris en compte pour la décision ;
- **Purement incrémental**, c'est-à-dire qu'il ne dispose pas de données *a priori* sur les utilisateurs et qu'il soit synchrone ;
- **Générique**, c'est-à-dire qu'il doit pouvoir être utilisé avec différents types d'éléments (par exemple, achat d'une voiture, d'une maison, d'un film) et cela de manière naturelle, c'est-à-dire, sans nécessité de traitements supplémentaires.

Deuxièmement, ce travail consiste à définir des techniques qui permettent d'apporter des parties de solutions à des verrous technologiques comme la désagrégation d'informations sur plusieurs critères et la prise en compte d'un nombre variable de critères dans le processus d'aide à la décision interactif, et ce sans définir au préalable de famille cohérente de critères sur laquelle est basée la décision.

Cette thèse se décompose en 3 parties principales, qui sont :

Premièrement, l'état de l'art, où nous allons définir tous les concepts utiles et nécessaires à la compréhension des mécanismes et enjeux mis en œuvre. Ainsi seront décrits les systèmes de recommandation, des méthodes d'aide à la décision, et des méthodes de prise en compte et évolution de profils/préférences. Les systèmes de Recommandation font partie des systèmes de Filtrage d'Information. Ils sont souvent adaptatifs, et ils mettent en œuvre des profils utilisateurs ; leur but étant, comme leur nom l'indique, de proposer à l'utilisateur des éléments qui puissent le satisfaire. Des profils utilisateurs peuvent être définis à travers plusieurs dimensions, comme les informations personnelles (âge, sexe, etc.) ou des informations à propos des domaines d'intérêts. Ensuite, nous avons décrit des méthodes d'aide à la décision et des méthodes issues de la théorie Multi-Attributs. Les critères ou attributs associés sont définis, et en particulier la différence entre ces deux termes. Nous avons expliqué le fonctionnement d'un système interactif d'aide à la décision, nous utilisons ce fonctionnement dans cette thèse, afin d'alterner entre des phases où l'on récupère de l'information à partir des actions utilisateurs, et des phases où l'on met à jour le profil de l'utilisateur en fonction de ces données. Enfin, nous avons exposé des méthodes permettant d'effectuer un apprentissage et une évolution des profils/préférences des utilisateurs. Ainsi, plusieurs opérateurs de désagrégation sont examinés, ceux-ci servant à décomposer des informations au sujet d'un élément, et d'utiliser ces informations pour mettre à jour les profils utilisateurs. L'opération d'agrégation correspond à l'opération inverse, ainsi son objectif est de rassembler des informations afin de déterminer une valeur correspondant à l'intérêt qu'à l'utilisateur pour un élément. De plus, des méthodes sont précisées afin de permettre de prendre en compte les interactions entre les critères, car dans la majorité des systèmes de recommandations, les critères sont considérés comme étant indépendants entre eux.

Les contributions que nous avons réalisées sont décrites dans la partie 2. Nous commençons par décrire les contraintes qui sont présentes, et celles que nous nous sommes

imposées, ainsi que les méthodes que nous avons utilisées ou définies afin de résoudre toutes ces contraintes. Pour cela, le modèle proposé dans cette thèse est décomposé et expliqué. De plus, les choix effectués sont justifiés. Dans ce modèle, nous examinons le système de recommandation proposé, le type de profil utilisé, la méthode d'apprentissage et d'évolution des préférences, ainsi que le mécanisme global de notre système. Ensuite, nous définissons la méthode employée pour initialiser notre système, pour ordonner les éléments, ainsi que pour calculer le score de pertinence de chaque élément pour l'utilisateur à un moment précis. Des méthodes de désagrégation de l'information sur plusieurs critères sont présentées. Enfin, sont exposées des méthodes définies au sein de cette thèse permettant de prendre en compte les interactions entre les critères, ainsi que leur mise en œuvre. Cette mise en œuvre utilisant un système tiers pour effectuer les calculs. Deux méthodes sont proposées, l'une utilisant de manière directe les informations obtenues à partir du système tiers sans utiliser les méthodes de désagréations définies précédemment, et l'autre, utilisant ces informations en complément de celles obtenues de manière classique au sein de notre système.

Et enfin, nous présentons la partie concernant les expérimentations, la validation de notre travail. Nous commençons par définir les outils de mesures permettant d'évaluer la performance de notre système, et les tests mis en place qui permettent d'évaluer notre système dans plusieurs cadres. Nous détaillons ensuite la méthodologie de nombreux outils permettant de mesurer, par exemple, la précision d'une prédiction de note, la prévision d'utilisation d'un élément, ou enfin, un ordre en utilisant des corrélations entre des éléments. Puis nous exposons le choix des outils que nous utilisons afin de tester notre système. Puis, nous détaillons les différents cadres et méthodologies utilisées afin de tester et valider notre système. Pour cela, nous commençons par comparer notre système avec celui d'Amazon, puis nous comparons les performances obtenues par nos deux algorithmes de désagrégation sur une base de données issue de l'utilisation par de vrais utilisateurs d'une plateforme en ligne proposant des films. Nous commençons par les tester avec une quantité limitée de critère pris en compte, puis en prenant en compte tous les critères disponibles. Ensuite, nous exposons les performances de notre système lorsqu'il est utilisé directement, par de vrais utilisateurs, à partir d'une plateforme en ligne. Enfin, nous comparons les performances de notre système avec celles d'un système Multi-Agents, d'un système de recommandation collaborative et en particulier une recommandation collaborative qualitative, et également avec un système utilisant un mélange de recommandation collaborative (n'utilisant que les actions des autres utilisateurs) et de recommandation sur le contenu (n'utilisant que les caractéristiques des éléments). Cela nous permet de tester de manière étendue notre système dans plusieurs cadres afin de pouvoir valider ses performances et son efficacité.

Partie I. État de l'art

Cette partie consiste à définir tous les termes et concepts utilisés au sein de cette thèse, ainsi que de détailler les avantages et inconvénients de chacun des principes ou méthodes définis.

Chapitre I. Systèmes de recommandation

Les systèmes de recommandation servent à sélectionner des objets/solutions adapté(e)s aux utilisateurs en fonction de leurs profils. Les utilisateurs n'ont plus le temps de regarder toutes les informations qui sont à leur disposition, c'est pour cela que les systèmes de recommandations peuvent se révéler extrêmement utiles, pour ordonner et filtrer les informations auxquelles les utilisateurs ont accès.

C'est ainsi que les systèmes de recommandations sont apparus afin de faciliter la prise de décision des utilisateurs lorsqu'ils doivent par exemple choisir entre plusieurs solutions.

Ces systèmes se basent sur des profils des utilisateurs représentant leurs préférences, et ce afin de filtrer, ordonner les informations pour l'utilisateur et sélectionner ce qui correspond le plus à sa demande.

I.1. Profils utilisateurs

Un profil peut être modélisé par différents types d'information permettant de caractériser un utilisateur ou un groupe d'utilisateurs. Ces types d'informations sont définis selon le contexte dans lequel le profil est utilisé. On peut y trouver par exemple des informations sur ses centres d'intérêt, ses préférences, des informations personnelles sur l'utilisateur, etc. Différentes formes de construction d'un profil ont été proposées dans la littérature. Un profil peut être construit de façon explicite à travers une liste de termes valués établie par l'utilisateur, de façon supervisée par le système en recueillant les jugements de l'utilisateur sur les items déjà reçus ou d'une façon implicite par observation du comportement de l'utilisateur lors de ses interactions avec le système.

I.1.A. Définition d'un profil

Afin de décrire les profils, nous nous basons sur la description d'un profil effectué dans le cadre du projet APMD (Accès Personnalisé à des Masses de Données) [BOUZ 05].

Un des objectifs du projet APMD est d'étudier les différents types d'informations constituant le profil utilisateur ainsi que les techniques de leur exploitation dans un environnement BD (Base de données) ou RI (Recherche d'Information). Dans ce contexte, il est indispensable de disposer d'une infrastructure générique de définition et de gestion de profils. Le profil utilisateur est représenté par six dimensions ouvertes capables d'accueillir la plupart des informations décrivant l'utilisateur et ses préférences :

- La dimension « Données personnelles » comprend toutes les informations caractéristiques de l'utilisateur qui sont relativement stables dans le temps, par exemple : nom, prénom, sexe, date de naissance, langues parlées, handicaps, etc.
- La dimension « Domaine d'intérêt » correspond aux critères dits « de contenu » : elle exprime les caractéristiques générales des informations que l'utilisateur souhaite obtenir du système, c'est ce qu'on utilise pour exploiter les préférences des utilisateurs
- La dimension « Préférences de livraison » couvre les contraintes souples intervenant entre la production des informations et leur mise à disposition à l'utilisateur, par exemple les dimensions de son écran. Également le moyen par lequel l'utilisateur accède à l'information, journal, télévision, internet, etc.
- La dimension « Sécurité » comprend toutes les informations liées aux privilèges ou au secret concernant les données, l'utilisateur ou les processus de traitement.
- La dimension « Historique des interactions » contient des préférences manifestées implicitement (sélection d'un élément) ou explicitement (mettre une note par exemple) lors de recherches d'informations passées, au travers des interactions avec le système. C'est grâce à cet historique que l'on peut construire un profil évolutif dans le temps.
- La dimension « Qualité » regroupe les exigences de qualité de l'utilisateur relatives aux informations, aux sources fournissant les informations et aux traitements appliqués à ces informations. Elle permet d'exprimer des préférences extrinsèques

sur l'origine de l'information, sa précision, sa fraîcheur, sa durée de validité, le temps nécessaire pour la produire ou la crédibilité de sa source.

Les préférences de l'utilisateur peuvent être exprimées de différentes manières selon les besoins des applications qui les utilisent. Elles peuvent concerner un facteur particulier ou un ensemble de facteurs.

La représentation la plus courante des préférences est la spécification des valeurs que l'utilisateur souhaite obtenir. Ceci est fait généralement par des semi-intervalles (seuils d'indifférence), en spécifiant une borne supérieure ou inférieure (ex. précision > 60%), ou par des ensembles flous (en donnant la fonction d'appartenance).

Une deuxième approche d'expression de préférences est la définition de relations d'ordre. Une relation d'ordre peut être donnée explicitement (ex. « j'ai plus confiance dans la source S1 que dans S2 ») ou définie implicitement par des fonctions permettant de dériver cet ordre (ex. « je préfère une source S_i à une autre S_j si S_i a une plus grande précision »). Dans ce dernier cas, l'ordre peut être établi selon un ou plusieurs facteurs.

Un autre type de préférences concerne la combinaison (agrégation) de facteurs. Cette préférence est exprimée par l'attribution d'un score à chaque facteur pour faire une somme évaluée. Ainsi l'utilisateur peut déterminer le degré d'importance de chaque facteur dans la méthode d'évaluation. Une alternative à cette approche est d'appliquer les préférences selon un ordre prédéfini (ex. on préfère les éléments qui ont une meilleure fraîcheur et en cas d'égalité ceux qui ont une meilleure exactitude). Finalement, des expressions plus complexes peuvent être utilisées pour décrire les préférences de qualité.

Une extension naturelle de systèmes utilisant un ou plusieurs profils est le cas des assistants personnels.

I.1.B. Profils pour assistants personnels

Les profils pour assistant personnel sont utilisés pour obtenir des périphériques personnalisés en fonction de leur utilisateur, de façon à améliorer l'expérience de celui-ci lors de l'utilisation dudit périphérique.

La plupart des systèmes effectuent leur adaptation en construisant un « profil de l'utilisateur » par apprentissage à partir des documents consultés. Ces systèmes utilisent en

effet, une base de données d'apprentissage qu'ils exploitent auparavant afin de définir le profil initial de l'utilisateur. Ce profil qui s'appuie généralement sur le Vector Space Model [SAL et al. 75], et est constitué d'un ou de plusieurs vecteurs définis dans un espace de termes ; les coordonnées correspondent aux scores associés aux termes retenus dans le profil (vecteurs de termes pondérés).

Définition d'un vecteur de termes pondéré [GOW 03] :

Pour un ensemble d'objets $O = \{o_1, o_2, o_3, \dots, o_n\}$ avec $n \geq 1$ et $n \in \mathbb{N}$, et $\{t_1, t_2, t_3, \dots, t_m\}$ avec $m \geq 1$ et $m \in \mathbb{N}$, un ensemble de tous les termes contenus dans chaque objet, termes décrivant leurs caractéristiques. Chaque objet o_j est représenté par un vecteur dans un espace à m -dimensions. Donc $o_j = [p_{j_{t_1}}, p_{j_{t_2}}, \dots, p_{j_{t_m}}]$, où $p_{j_{t_i}}$ représente le poids du terme t_i dans l'objet o_j .

L'utilisation de plusieurs vecteurs correspond à deux préoccupations : pouvoir prendre en compte des centres d'intérêt multiples, de définir des profils génériques vis-à-vis d'un utilisateur (on parle alors de meta-profils) et gérer leur évolution dans le temps. Dans ce dernier cas, le système utilise généralement deux groupes de vecteurs, dont l'évolution est gouvernée par des règles différentes (c'est le cas d'Alipes et de News Dude [BIL 99]).

Exemple :

Avec des couples de type : (attributs, score reflétant la préférence pour l'utilisateur)

Profil photographie = {(photographie, 8), (art, 5), (couleur, 3), (caméra, 7)}

Profil voiture = {(voiture, 9), (moteur, 2), (puissance, 7), (course, 8), (couleur, 6)}

On remarquera que l'on utilise ici des scores et non des poids, et que les scores pour un même attribut peuvent varier entre 2 profils. L'utilisation de score permet en effet une plus grande amplitude dans les écarts des préférences des utilisateurs et permet une plus grande réactivité aux changements de préférences des utilisateurs.

Dans la majorité des cas, l'évolution des profils repose sur des variantes de la méthode de « bouclage de pertinence » [ROC 71]. L'évaluation des éléments retrouvés peut être explicitement fournie par l'utilisateur (une note), ou être déduite, à partir de l'observation de l'activité de l'utilisateur. Différents indicateurs sont utilisables tels que : les « raw data » ou données brutes (mouvements et clics de souris, les temps passés sur la page), le nombre de liens activés, la sélection d'un élément dans une liste, etc.

L'ajustement du score des termes fait de plus en plus appel à des techniques d'apprentissage, comme des réseaux de neurones, des probabilités Bayésiennes, des

algorithmes à base de règles, des algorithmes génétiques et plus récemment de l'apprentissage par renforcement.

La construction d'une ontologie personnelle [HUN 99] offre une alternative intéressante aux approches présentées ci-dessus. Une ontologie est une description (comme une spécification formelle d'un programme) des concepts et des relations qui peuvent exister pour un agent (par exemple, une personne) ou d'une communauté de l'agent. L'ontologie constitue en soi un modèle de données représentatif d'un ensemble de concepts dans un domaine, ainsi que des relations entre ces concepts. Elle est employée pour raisonner à propos des objets du domaine concerné. Plus simplement, on peut aussi dire que « l'ontologie est aux données ce que la grammaire est au langage ».

« Ainsi dans le système OBIWAN [GAU et al. 03] il est proposé une approche dans laquelle la personnalisation est fondée sur l'élaboration d'une ontologie personnelle, obtenue par la sélection dans une ontologie générale de nœuds estimés correspondant aux intérêts de l'utilisateur. Cette sélection est opérée sur la base d'un ensemble de documents organisé par l'utilisateur, le résultat étant une hiérarchie de concepts. La construction d'une ontologie personnelle peut également être réalisée à partir des techniques employées par les systèmes élaborant automatiquement des thésaurus à partir de l'analyse de collections de documents, dans le but de les exploiter pour étendre ou interpréter des requêtes. Bien que généralement appliquée dans le contexte de recherche sur des collections connues et finies, cette approche peut être adaptée au domaine du Web si une sélection de documents est fournie par l'utilisateur. » [BOT et al. 03]

L'idée de la mise en place d'une ontologie personnelle est très tentante, car ainsi on dispose d'un profil très précis et très personnel vis-à-vis de l'utilisateur. Cependant, cela oblige l'utilisateur à un très grand investissement afin de définir, par exemple, les concepts et les relations entre ces concepts. Ce qui est très contraignant et est par le fait très peu utilisé.

Nous pouvons ainsi construire des profils adaptés et nous en servir dans un système de recommandation afin d'effectuer des suggestions à l'utilisateur. Ces systèmes de recommandations font partie du domaine du filtrage d'information que nous allons décrire.

I.2. Filtrage d'informations

Le Filtrage d'Information (FI), sélectionne des éléments provenant de sources, souvent non connues *a priori* [TEB 04]. Le filtrage d'information est un processus dual de la recherche d'informations [BEL 92]. Ceci traduit qu'un processus de recherche peut simuler un processus de filtrage. L'architecture générale d'un système de filtrage d'informations est semblable à celle d'un système de recherche d'informations. Dans le cas qui nous intéresse, nous ne mettons pas en œuvre uniquement des documents (la majorité des systèmes de filtrage d'information utilisent les documents), c'est pour cela que l'on parle ici, d'objets, qui peuvent être de nature différente d'un document (par exemple : voiture, maison, *etc.*)

I.2.A. Architecture générale d'un système de filtrage d'informations

Le processus général consiste à représenter le besoin en information, en un objet, de l'utilisateur, et en parallèle les objets de la source de données, afin de déterminer l'appariement entre chaque objet et le profil puis de décider si l'objet est pertinent (est intéressant pour l'utilisateur). L'adaptation est un processus de base en filtrage d'information, elle concerne le profil (voir I.1.A) et éventuellement d'autres informations qui peuvent définir le profil tel que des statistiques sur les activités de l'utilisateur, des éléments que l'utilisateur a aimés, *etc.*

À ce titre, la plupart des systèmes de filtrage d'information sont basés sur des modèles de recherche d'informations. Ainsi, les éléments et les profils sont représentés par des listes de critères pondérés. L'appariement objet-profil consiste à mesurer une similarité et cela grâce à des mesures de distance ou d'agrégation de données.

Or, en l'absence de base d'apprentissage de référence, la détermination des pondérations adéquates associées aux profils et aux objets ou encore leurs caractéristiques (facteurs) sont les problèmes majeurs rencontrés dans ce domaine. « En effet, dans un système de recherche d'informations, les techniques de pondération et de reformulation automatique de requêtes, basées sur la collection entière d'objets, se sont avérées efficaces, or, dans un système de filtrage d'information, à l'initialisation du processus de filtrage, on ne dispose d'aucune connaissance sur les éléments à filtrer pour pouvoir construire une fonction de décision, ni pour bien pondérer les profils et les éléments entrant. » [TMA 02].

De plus, l'adaptation des profils, leur mise à jour pose un problème. En effet il faut savoir à quel moment interagir avec eux afin de les corriger.

Les solutions proposées aujourd'hui sont les suivantes :

- solution synchrone : adaptative, les facteurs sont déduits à partir des objets rencontrés/filtrés cumulés dans le temps,
- solution asynchrone ou différée : les facteurs sont déduits à partir de collections d'objets existantes. Ces collections fournissent également des exemples d'éléments pertinents pour chaque profil.

La solution synchrone offre la possibilité d'avoir un système qui évolue en temps réel. Ainsi l'utilisateur a un retour direct de ses actions et réponses, afin par exemple de reconstruire/faire évoluer son profil. Cependant, cette solution oblige à utiliser des calculs adaptés dans leur complexité et coût/temps de calcul par rapport à la réactivité du système voulu. Ou alors, pour pallier ce problème, investir de manière importante dans des systèmes de calculs performants. Donc, soit il faut des calculs peu gourmands en ressources, soit il faut faire de lourds investissements.

La solution asynchrone permet de se passer des problèmes de coût de calcul et de réduire les investissements, mais il en ressort une expérience utilisateur réduite dans sa qualité. En effet, les systèmes asynchrones mettent à jour le profil des utilisateurs et effectuent divers autres calculs de manière différée. Cependant, les utilisateurs ont une influence de leurs actions et réponses sur leur interface, et dans les objets qui leur sont proposés, de manière décalée. De plus, de tels systèmes peuvent même conduire à des résultats, dans les objets proposés, qui ne correspondent pas aux préférences de l'utilisateur qui ont éventuellement changé avant la mise à jour en différé de leur profil.

De nombreux systèmes de filtrage d'informations mettent en œuvre une fonction dite de décision pour accepter ou rejeter un élément.

I.2.B. Fonction de décision

La notion de fonction de décision n'existe pas dans le domaine de la recherche d'informations. Elle est spécifique aux systèmes de filtrage d'information et elle est souvent de type « seuil ». On parle aussi de fonction de seuillage. L'objectif de cette fonction est de décider d'accepter ou de rejeter un élément. Dans cette étude, nous n'utilisons pas de fonction de décision (de seuillage) pour la simple raison que l'on ne refuse aucun objet,

cependant des informations peuvent, pour des raisons d'ergonomie, ne pas être affichées à l'utilisateur.

I.2.C. Le filtrage adaptatif incrémental

La plupart des systèmes de filtrage existants se basent sur des modèles de recherche d'informations.

« Dans ces systèmes, la notion d'incrémentalité est très peu respectée, car :

- Ils utilisent souvent des bases d'entraînement [ZHAI et al. 98] [ROB 99]. Une base d'entraînement correspond à une base de données d'apprentissage utilisée pour initialiser le système. Ces systèmes se basent sur les statistiques des termes et des documents dans ces bases pour estimer les valeurs de plusieurs paramètres de filtrage. Or dans le vrai cas de filtrage, ces statistiques sont très variables au cours du filtrage et inconnues à l'initialisation du processus. De plus, les documents issus des bases utilisées dans l'entraînement peuvent avoir des caractéristiques différentes des documents provenant de la source, les statistiques utilisées dans l'entraînement peuvent alors être peu crédibles.
- Ils effectuent l'apprentissage du profil d'une manière quasi différée [KWO et al. 00] [HOA et al. 99]. En effet, plusieurs effectuent l'apprentissage de profil à chaque réception de n (10, 100, 1000) documents. Avec cette manière périodique de fonctionnement, le système perd beaucoup de qualités incrémentales et tend vers les systèmes de filtrage différé.
- Plusieurs informations sont parfois supposées connues avant le démarrage du filtrage (proportion de documents pertinents, quelques documents pertinents d'entraînement...) [ZHAI et al. 98], ceci n'est pas vrai dans le cas du filtrage, car pour démarrer le système ne nécessite aucune information sauf, éventuellement, le profil initial de l'utilisateur.

» [TMA 02]

L'évaluation des systèmes de filtrage d'information [HUL 97] [HUL 98] [ROB 00] [Su 92] est effectuée en utilisant d'autres procédures et mesures que celles utilisées en Recherche d'Information : les systèmes de filtrage d'information doivent accepter ou rejeter chaque élément en arrivée, par conséquent le résultat de filtrage est traité comme une liste non ordonnée de documents.

Or nous cherchons dans notre système à ordonner des objets en fonction du profil de l'utilisateur en sortie du processus, et c'est une des possibilités offertes par les systèmes de recommandations. En effet, ces systèmes permettent d'ordonner les éléments qui sont pertinents pour l'utilisateur, ils ne se limitent pas à décider si un élément sera pertinent ou non et permettent aussi de déterminer à quel point il est pertinent et pourquoi.

I.3. Systèmes de recommandation

Un système de recommandation utilise des informations implicites ou explicites obtenues d'un utilisateur pour lui suggérer de nouveaux éléments qu'il pourrait aimer. Il utilise des mécanismes qui lui permettent de déterminer ce qui est important pour l'utilisateur. Ainsi il utilise ses informations pour assister l'utilisateur en s'adaptant à lui et en lui suggérant des solutions à ses demandes.

Bien que les systèmes de recommandations aient été utilisés quelques fois dans le domaine de l'Intelligence Artificielle (IA), leurs domaines d'applications et d'utilisations n'ont que récemment augmenté, cela grâce à l'explosion des systèmes de données électroniques qui ont inondé les systèmes ces dernières années.

Le système apprend de manière classique en fonction d'éléments qui ont été aimés ou non. Un système de recommandation typique se décompose en 3 étapes :

1. L'utilisateur fournit une liste d'exemples de ses goûts, qui peuvent être explicites, comme des notes sur des éléments spécifiques, ou implicites, comme des URL visitées, ou une sélection d'un objet parmi une liste qui lui est présentée,
2. Ces données sont utilisées par le système afin de créer une représentation du profil de l'utilisateur, ce qu'il aime et ce qu'il n'aime pas,
3. Le système calcule des recommandations en fonction du profil de l'utilisateur.

Afin de réaliser ce genre de système, plusieurs méthodes sont apparues pour recommander des éléments à l'utilisateur. C'est ainsi que de nombreux systèmes de recommandation ont été créés, pour, en particulier, classifier des textes et en recommander ensuite à l'utilisateur. La plupart utilisent un système de « bag-of-words Naive Bayesian text classifier » [MOO 00]. Dans ce modèle, un texte (comme dans un document) est représenté comme une collection non ordonnée de mots, sans prendre en compte la grammaire, ni l'ordre même des mots. On utilise ensuite des techniques comme TF-IDF [RAM 03], afin de construire le profil de l'utilisateur, et donc par la suite d'utiliser ce profil afin de produire des suggestions, ou recommandations.

La plupart des systèmes de recommandation utilisent un système de personnalisation basé sur le passé de l'individu. Cette personnalisation utilise par exemple, l'historique de navigation de l'utilisateur ou son historique d'achat. Et cela aussi bien pour déterminer ce que l'utilisateur aime, ou ce qu'il n'aime pas.

Afin d'améliorer les performances des systèmes de recommandations, des techniques sont venues enrichir cette méthode. En effet, ce système de personnalisation ne prend en compte que des données précédemment fournies par l'utilisateur. Or, de nombreuses autres informations sont utilisables, telles que la distance entre les caractéristiques de deux objets, ou encore ce que font les autres utilisateurs. C'est ainsi que sont nées les méthodes de recommandation collaborative et de recommandation sur le contenu.

I.3.A. Recommandation collaborative (collaborative filtering)

La recommandation collaborative ou recommandation sociale utilise des connaissances à propos des autres utilisateurs afin d'effectuer des recommandations.

Pour cette recommandation, l'hypothèse sous-jacente est que des utilisateurs similaires aimeront les mêmes objets. Des exemples de systèmes de recommandation collaborative peuvent être trouvés dans [BIL 98], [BREE et al. 98].

La recommandation collaborative peut être utilisée dans des domaines où il n'y a pas beaucoup de contenu associé à des éléments, ou lorsque le contenu est difficile à analyser pour un ordinateur, par exemple des idées, des opinions, etc. Deuxièmement, un système de recommandation collaborative a la capacité de fournir des recommandations fortuites. Il peut recommander des objets qui sont pertinents pour l'utilisateur, mais ne contiennent pas de contenu commun avec le profil de l'utilisateur. On remarque également que lorsqu'un nouvel utilisateur qui a peu utilisé le système l'utilise, il obtient tout de même de bonnes recommandations, car le système le compare aux autres profils, plus riches en informations, et lui recommande rapidement du contenu approprié.

Le principe consiste à recommander des objets sur la base du comportement passé des utilisateurs similaires. On effectue un calcul de similitudes et de différences entre les utilisateurs, pour cela plusieurs méthodes ont été mises au point. Par exemple, le concept d'algorithme des « K plus proche voisin » a été inclus dans la mise en œuvre des systèmes de recommandation [RES et al. 94]. Et ensuite, on exploite les données obtenues afin de réaliser la recommandation.

Définition de la méthode des « K plus proches voisins » :

L'algorithme des K plus proches voisins est un des algorithmes de classification. Le seul outil dont on a besoin est une distance entre les éléments que l'on veut classer. Si l'on représente ces éléments par des vecteurs de coordonnées, il y a en général plusieurs choix possibles pour ces distances. Par exemple, la distance euclidienne, ou d'autres mettant en œuvre des mesures plus sophistiquées pour tenir compte, si nécessaire, de paramètres non numériques.

Le fonctionnement de cette méthode est décrit comme ceci : dès que l'on reçoit un nouvel élément, on calcule sa distance à tous les éléments composant notre base de données. Si cette base comporte 100 éléments, alors on calcule 100 distances et on obtient donc 100 nombres réels. Si $k = 25$ par exemple, on cherche alors les 25 éléments ayant la distance la plus réduite. Ces 25 nombres correspondent donc aux 25 éléments de la base qui sont les plus proches de l'élément.

Il y a cependant plusieurs inconvénients au filtrage collaboratif. Tout d'abord, pour l'extraction de relations, de nouveaux objets non encore classés ou non encore marqués peuvent être abandonnés lors du processus de recommandation, ainsi les nouveaux objets ne sont pas recommandés. Le deuxième problème est que le filtrage collaboratif peut ne pas couvrir les cas extrêmes. Si les échelles des critères dans les profils d'utilisateurs sont de petites tailles ou si les utilisateurs ont des goûts rares ou uniques, les calculs de similarité entre les utilisateurs ne donneront pas de bons résultats, et ainsi les recommandations faites à l'utilisateur ne seront pas correctes. En effet, de nombreux systèmes utilisent le filtrage collaboratif, ces systèmes peuvent satisfaire un grand nombre de personnes qui n'ont pas des goûts très différents. Ils ne peuvent pas satisfaire les utilisateurs avec des goûts précis qui ne sont pas très populaires, et ils n'offrent pas de bons résultats avec peu d'utilisateurs [CHE 06]. Le troisième problème est la fréquence de mise à jour. Si de nouvelles informations des utilisateurs doivent être incluses dans le processus de recommandation en temps réel, la latence des données va augmenter le temps d'attente pour le résultat de la requête, et ainsi rendre le système peu réactif. Plus d'informations à ce propos peuvent être trouvées dans [ADO 05].

I.3.B. Recommandation sur le contenu (content based)

La recommandation sur le contenu utilise uniquement les informations obtenues à propos d'un objet.

Pour cette recommandation, l'hypothèse sous-jacente est de comparer les objets entre eux, et de garder une matrice des relations détectées, pour ainsi proposer un objet proche d'un objet apprécié par l'utilisateur. Des exemples de systèmes utilisant ce principe peuvent être trouvés dans [PAZ 97], [ZHAN et al. 02].

Le principe consiste donc à recommander des objets en se basant uniquement sur les qualités et propriétés intrinsèques de l'objet lui-même et en les corrélant avec les préférences et intérêts de l'utilisateur. Ainsi par un calcul de distance on obtient une matrice de relation des objets les uns par rapport aux autres. Cette matrice est calculée en différé et mise à jour lors de l'ajout de nouveaux objets dans le système.

Les avantages de ces systèmes sont qu'ils permettent de proposer des objets à l'utilisateur sans avoir d'influence des autres utilisateurs. En effet, le fait de ne se baser que sur les objets permet ainsi d'avoir des recommandations efficaces sur de naissantes ou petites plateformes. Ceci étant dû à l'absence de nécessité de posséder une base de données contenant les actions des autres utilisateurs. Lorsqu'un nouvel objet est ajouté dans le système, il est très rapidement recommandé aux utilisateurs ayant auparavant apprécié un objet similaire. Il permet, de plus, de satisfaire tous les utilisateurs, en effet ce type de système ne souffre pas des problèmes que présentent les systèmes collaboratifs vis-à-vis des profils atypiques. En effet, ces systèmes offrent de bons résultats même avec des profils utilisateurs rares ou uniques. Cela permet ainsi d'avoir des profils utilisateurs très personnalisés et/ou très différents de celui des autres utilisateurs, et de pourtant pouvoir fournir de bonnes recommandations. On peut également remarquer que ce genre de système va recommander des objets similaires à ceux que l'utilisateur apprécie très rapidement et dès que l'utilisateur a dans son profil au moins un objet, ainsi les nouveaux utilisateurs ont des recommandations correctes très rapidement.

Mais il faut reconnaître que ces systèmes souffrent également de problèmes. En effet, la majorité des utilisateurs ont des profils assez similaires (ou peuvent être rassemblés dans des catégories fixes) et la recommandation collaborative donne de très bons résultats dans ces cas-là, résultats souvent meilleurs. Mais le problème majeur reste la description des objets, en effet tout ou presque dépend de leurs caractéristiques déclarées, ainsi les objets peu renseignés dans leur description se verront assez peu recommandés. On remarque également que les utilisateurs ne se verront proposer que des objets similaires à ceux qu'ils connaissent déjà. Plus d'informations à ce propos peuvent être trouvées dans [ADO 05].

I.3.C. Recommandation hybride

La recommandation hybride utilise à la fois la recommandation sur le contenu et à la fois la recommandation collaborative. Elle permet d'utiliser un système pour pallier aux défauts de l'autre, des exemples peuvent être trouvés dans [BAS et al., 98], [SCHE et al. 02].

Parmi les systèmes de recommandation les plus célèbres aujourd'hui, on trouve le système de recommandation d'Amazon.com [LIND et al. 03]. Amazon utilise les deux approches (sociale et objet). Amazon possède un système très sophistiqué. Les recommandations sont d'une part personnalisées en se basant sur le comportement individuel passé de l'utilisateur (historique de navigation et historique d'achat), et d'autre part Amazon utilise aussi les caractéristiques de l'article lui-même (recommandation objet) et les comportements d'autres personnes (recommandation sociale).

De nombreuses méthodes, algorithmes et modèles ont été proposés pour résoudre le problème de similarité dans les systèmes de recommandation collaborative de filtrage. Une des méthodes les plus courantes afin de déterminer la similitude est le calcul de cosinus d'angle. Le système de recommandation d'Amazon utilise cette mesure pour décider de la similitude entre chaque paire d'articles achetés par chaque client. Cette similitude servant à établir la matrice contenant les relations objet-objet. On a de cette manière une matrice définissant la distance entre chaque objet, mais aussi entre chaque utilisateur. Ainsi dans le système d'Amazon, on utilise à la fois la recommandation collaborative, et à la fois la recommandation sur le contenu. Tous ceux qui ont déjà fait un achat sur Amazon ont probablement dû lire le message de la part d'Amazon qui nous dit « les gens qui ont acheté x ont aussi acheté y ». Cette approche est tout simplement l'approche « *item-based nearest neighbor* » (*item-centric* ou *item-based collaborative filtering*) [SCH et al. 07].

Pour donner un ordre d'idée de l'importance de ce système, on remarquera que la valeur du système de recommandation compte comme représentant un tiers de la valeur d'Amazon en 2009.

Cependant, il existe d'autres systèmes de recommandations, qui sont moins utilisés en raison de leurs caractéristiques.

I.3.D. Autres systèmes de recommandations

Par ailleurs, d'autres types de systèmes de recommandation ont également été proposés dans la littérature. Par exemple, Burke [BUR 02] distingue les suivants (en plus de ceux décrits ci-dessus) :

- La recommandation démographique, qui classe les utilisateurs selon les attributs de leur profil personnel, et effectue des recommandations fondées sur des catégories démographiques. Un des premiers exemples est le système de stéréotype basé sur Grundy, qui a été développé pour soutenir les recherches de livres dans une bibliothèque [REI 83].
- La recommandation basée sur une valeur d'utilité, qui effectue des suggestions basées sur un calcul de l'utilité (indice d'intérêt) de chaque élément pour un utilisateur, et pour laquelle une fonction d'utilité doit être stockée. Des exemples peuvent être trouvés dans [HUA 08].
- Des recommandations basées sur l'exploitation de connaissance, qui propose des articles en effectuant des inférences logiques grâce aux préférences des utilisateurs. Une représentation des connaissances (par exemple, par règles) sur la façon dont un élément satisfait un besoin d'un utilisateur en particulier est nécessaire. Un exemple peut être trouvé dans [BUR 00].

I.4. Conclusion

Nous possédons à présent une proposition de formalisation des profils utilisateurs, ces profils étant utilisés à travers diverses méthodes et en particulier les systèmes de recommandations. Les systèmes de recommandation faisant partie des systèmes de filtrage d'informations, la plupart du temps ils viennent en complément de systèmes de Recherche d'Informations. On remarquera que plusieurs types de systèmes de recommandation existent, et qu'ils utilisent des approches/concepts très différent(e)s. Ces différences de concepts engendrent la conception de nombreuses méthodes venant s'appuyer sur ces concepts.

Voici un tableau récapitulatif de ces systèmes de recommandation assortis de leurs principaux avantages et inconvénients.

	Avantages	Inconvénients
Recommandation collaborative	<ul style="list-style-type: none"> ● Efficace rapidement, et ce même si l'utilisateur a peu interagi avec la plateforme ● Utilisable avec des données difficilement analysables ● Fournit des recommandations fortuites 	<ul style="list-style-type: none"> ● Nécessite obligatoirement une importante base de données des comportements des utilisateurs (impossible pour une plateforme naissante) ● Nouveaux objets non recommandés ● Difficultés à gérer des profils peu populaires ● Problème de fréquence de mise à jour et de puissance de calcul
Recommandation sur le contenu	<ul style="list-style-type: none"> ● Efficace, et ce même avec des profils rare ou unique ● Indépendance par rapport aux autres utilisateurs ● Utilisable sans base de données des comportements des utilisateurs (plateforme naissante) ● Recommande les nouveaux objets ● Peu coûteux en puissance de calcul 	<ul style="list-style-type: none"> ● Très dépendant de la description des objets à recommander ● Recommandations pouvant frustrer l'utilisateur, car pouvant être trop semblables entre elles sans traitement supplémentaire
Recommandation hybride	<ul style="list-style-type: none"> ● Combinaison des avantages de la recommandation collaborative et sur le contenu ● Performance 	<ul style="list-style-type: none"> ● Combinaison des inconvénients de la recommandation collaborative et sur le contenu ● Nécessite des algorithmes mixant les résultats obtenus avec le système collaboratif

		et sur le contenu
Recommandation démographique	<ul style="list-style-type: none"> ● Utilise les données démographiques ● Classifie les utilisateurs 	<ul style="list-style-type: none"> ● Classification des utilisateurs, besoin d'une base homogène d'utilisateurs et importante ● Efficacité limitée, solution d'appoint uniquement (exemple : recherche de livres en bibliothèque)
Recommandation avec valeur d'utilité	<ul style="list-style-type: none"> ● Fonction d'utilité, fonctionnement semblable à la recommandation sur le contenu 	<ul style="list-style-type: none"> ● Évolution ● Recalcul complet à chaque nouvelle information) ● Impossibilité d'avoir des utilisateurs à rationalité limitée ● Complexité
Recommandation à partir de l'exploitation de connaissances	<ul style="list-style-type: none"> ● Génération de nouvelles préférences ● Liens entre les préférences mises en œuvre 	<ul style="list-style-type: none"> ● Impossibilité d'avoir des utilisateurs à rationalité limitée ● Exactitude des inférences ● Nécessite une représentation des connaissances de l'utilisateur

Tableau 1. *Récapitulatif de différents systèmes de recommandation*

Afin de mieux prendre en compte les préférences des utilisateurs, nous proposons d'utiliser des systèmes multicritère/multiattribut. Cela permet d'avoir une meilleure gestion et prise en compte des préférences de l'utilisateur.

Chapitre II. Aide à la Décision Multicritère / Multiattribut

L'aide à la décision Multicritère (MCDA) est utilisée afin d'apporter un support à la décision pour les utilisateurs. Il s'agit là d'assistance à la décision et non de remplacer la décision d'un décideur. Cette assistance est faite à travers l'obtention des préférences de l'utilisateur puis de la sélection de la solution la plus adaptée à l'utilisateur et à son problème. L'utilisation de plusieurs critères permet de prendre en compte les préférences de l'utilisateur à travers des critères, présents dans son profil, qui sont plus ou moins importants pour l'utilisateur.

Afin de bien situer la différence entre un attribut et un critère, il nous faut les définir :

- **Attribut** : *Caractéristique décrivant chaque objet (âge, diplôme, résultats aux tests d'aptitude, prétentions).*
- **Critère** : *Exprime les préférences du décideur relativement à un point de vue (ex : voiture puissante) ; Cette notion intègre la structure de préférence du décideur sur ce critère (maximiser la puissance, maximiser la vitesse). Un critère peut faire référence à un ou plusieurs attributs. Par exemple, le critère de « compétence », fait référence à des attributs, tels que les diplômes, les expériences passées, etc.*

Lorsqu'on introduit dans l'évaluation d'un attribut des caractères subjectifs, relatifs aux préférences du décideur, on utilise plus volontiers le terme « critère ».

Cependant, on peut souvent définir un critère par un seul attribut et ainsi définir l'approximation suivante : **attribut \simeq critère**. Pour reprendre le critère de « compétence », il peut être défini, par exemple, par un seul attribut correspondant à un indice de compétence calculé en fonction d'autres attributs.

C'est pourquoi, généralement et pour la suite, on utilise l'approximation suivante : **attribut \simeq critère**.

II.1. Aide à la Décision Multicritère

Il existe un très grand nombre de méthodes d'aide à la décision multicritère. La plupart d'entre elles sont rassemblées et expliquées par [ROY 85], [SCHA 85], [VIN 89], [POM et al. 93], [ROY 93] et [FIG et al. 05].

« Deux « écoles » existent dans le domaine MCDA et suivent des principes de base assez différents [MAY et al. 94]. La première est « l'école Américaine », qui utilise le plus souvent une fonction d'utilité additive qui permet de combiner les valeurs d'utilité en une note globale pour l'action. La méthode la plus simple de cette catégorie est celle de la somme pondérée, où la note globale d'une action est la somme pondérée des notes de chaque critère de sélection multipliée par le poids du critère. Les méthodes de ce type sont MAUT, Multi Attribute Utility Theory [KEE 76], MHM pour Méthode de hiérarchisation multicritère [SAA 84] ou AHP pour Analytical Hierarchy Process. » [FRA 96]

L'école Européenne favorise en général l'utilisation de méthodes fondées sur des notions de surclassement entre les actions potentielles. La notion de surclassement correspond à dire qu'un élément est préféré (par rapport à un ou plusieurs points de vue) à un autre. Cette notion est plus détaillée au chapitre II.1.A.

« Les méthodes les mieux connues sont les méthodes ELECTRE, Élimination Et Choix Traduisant la RÉalité [ROY, 68, 78], [ROY 71, 73], [ROY 82a, 82b], [ROY et al. 85], [ROY 91], [YU, 92a, 92b], [MAY et al. 94] et PROMETHEE, Preference Ranking Organisation METHod for Enrichment Evaluations [BRA et al. 84]. Les relations de surclassement permettent de construire le graphe de surclassement, à partir duquel est construit le classement final des actions potentielles. Ces méthodes permettent aussi d'établir des notions d'indifférence et d'incomparabilité entre les actions, ce qui correspond souvent le mieux à la façon de penser du décideur. Malheureusement ces deux notions soulèvent souvent des problèmes d'interprétation du graphe de surclassement. De nombreuses modifications aux méthodes de base ont été proposées, notamment pour les méthodes ELECTRE, ce qui en explique le grand nombre de variantes. Ces modifications sont liées à l'introduction de plusieurs seuils de surclassement et à l'utilisation de concepts flous. Ces modifications peuvent éviter la définition de fonctions d'utilité pour les critères de sélection, tout en permettant d'améliorer les performances techniques de la méthode. La méthode devient néanmoins souvent encore plus difficilement accessible au décideur. » [FRA 96]

Il existe encore un grand nombre d'autres méthodes qui n'appartiennent ni à l'une ni à l'autre des deux « écoles », comme QUALIFLEX [PAE 78]. Cette méthode tient compte du fait qu'il est davantage dans la nature humaine de donner un rang à des objets que de leur attribuer des cotes. Il y a également, par exemple, les méthodes utilisant des principes de coût et bénéfice [RAI, 68, 73] et [SCHL, 59, 69]. Toutes ces méthodes demandent, elles aussi, l'obtention a priori de poids et d'autres paramètres.

À présent que le cadre multicritère est posé, il nous faut définir les différentes sortes de critères existants.

II.1.A. Les critères

Définition : Un *critère* est une fonction $g : A \rightarrow X \subset \mathbb{R}$ qui permet, relativement à un point de vue donné et pour un acteur identifié, de comparer deux actions quelconques a et b , de telle manière que :

$$g(b) \geq g(a) \Rightarrow bSa$$

Où S surclasse : "est au moins aussi bon que" (nous supposons ici que g est un critère à maximiser.)

La manière dont s'effectue la comparaison des actions sur chaque critère dépend du pouvoir de discrimination de celui-ci [VIN 80], [ROY 85], [ROY 87]. Lorsque le pouvoir de discrimination de g est parfait, on dit que g est un vrai-critère. Mais si g possède un pouvoir de discrimination imparfait, il s'agira d'un quasi-critère ou d'un pseudo-critère.

Un critère est souvent utilisé de la façon suivante :

$$g(b) > g(a) \Leftrightarrow bPa \quad P : \text{préférence stricte}$$

$$g(b) = g(a) \Leftrightarrow bIa \quad I : \text{indifférence}$$

Pour nuancer la transition entre I et P , on introduit une zone intermédiaire appelée préférence faible et notée Q .

Ainsi g induit sur A un pré-ordre total, structure « agréable » pour déterminer un choix, un tri ou un classement. En effet, g permet de pouvoir comparer chaque élément 2 à 2, l'un par rapport l'autre, et permet ainsi de connaître lequel est préféré, et ce pour chaque couple d'actions.

II.1.A.i. Vrai critère

Dans un vrai critère, les seuils de préférences sont identiques et valent 0. Son pouvoir de discrimination est parfait : une faible différence, aussi minime soit-elle, fait basculer la situation d'indifférence dans une situation de préférence stricte. On peut le voir sur la figure suivante où on considère la performance d'actions (a et b) par rapport à un critère g .

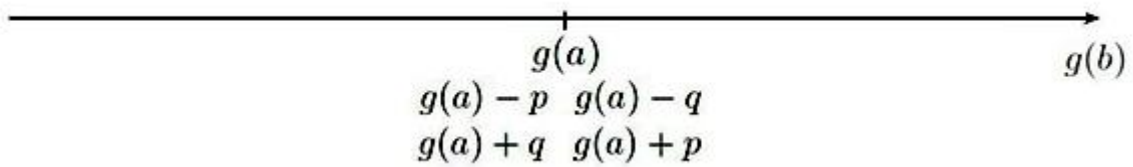


Figure 1. *Vrai critère*

Vrai critère avec q et p : seuil de préférence respectivement faible et fort, $q = p = 0$

Avec un vrai-critère (Figure 1), on ne peut que discriminer sans nuances des situations d'indifférence et des situations de préférence stricte et rejeter toute situation de préférence faible. C'est pour cette raison que l'on dira qu'un vrai-critère possède un pouvoir discriminant absolu.

II.1.A.ii. *Quasi-critère*

On admet donc à présent qu'une légère différence entre deux évaluations peut-être compatible avec une situation d'indifférence. À cet effet, on considère un nombre positif, noté q et appelé seuil d'indifférence.

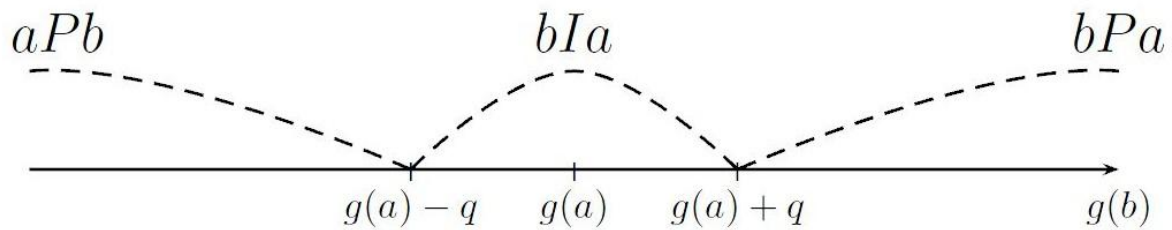


Figure 2. *Quasi-critère*

Quasi-critère avec q : seuil d'indifférence ($0 \leq q$)

Le pouvoir de discrimination de g est imparfait dans la mesure où la situation d'indifférence est associée non plus à un point (valeur 0), mais à un intervalle $[-q, q]$. Dans ce cas, on dit que g est un quasi-critère (Figure 2).

II.1.A.iii. Pseudo-critères

Dans le cas précédent, le passage de l'indifférence à la préférence stricte s'effectue sur la base d'un seuil fixé d'une manière précise : une différence $g(b) - g(a)$ tout juste plus grande que q conduit directement de bIa à bPa . Pour nuancer la transition entre I et P, on introduit une zone intermédiaire appelée préférence faible et notée Q.

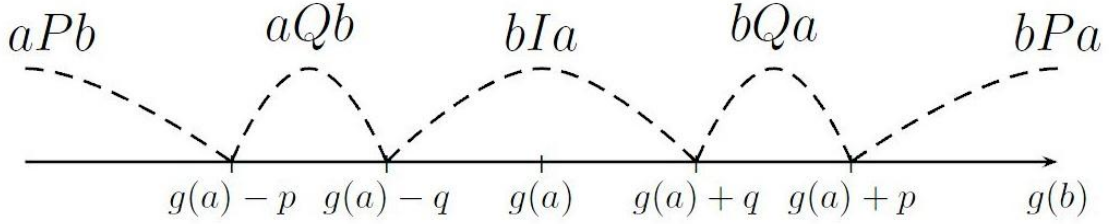


Figure 3. Pseudo-critère

Quasi-critère avec q : seuil d'indifférence ($0 \leq q$)

Pseudo-critère avec q et p : seuil de préférence, $0 \leq q \leq p$

L'introduction de ce seuil permet de définir une zone d'hésitation entre l'indifférence et la préférence stricte : c'est la zone de préférence faible (Figure 3). Par contre, l'indifférence n'aura lieu entre deux actions a et b que si leurs performances obtenues par le critère g sont égales : $g(a) = g(b)$ ou s'il existe un écart entre $g(b)$ et $g(a)$ inférieur au seuil de préférence faible pour a par rapport à b , c'est-à-dire que $-q < g(b) - g(a) < q$.

Si par contre cet écart est supérieur à p , c'est-à-dire si $g(b) - g(a) > p$ ou si $g(b) - g(a) < -p$, la relation liant a à b est une relation de préférence stricte.

Et si cet écart est compris entre q et p , c'est-à-dire que $q < g(b) - g(a) < p$, ou $-q > g(b) - g(a) > -p$, alors la relation est une relation de préférence faible.

II.1.A.iv. Incomparabilité

On remarquera que : si deux éléments a et b diffèrent sur presque tous leurs critères sans que l'un d'entre eux ne l'emporte pour une coalition importante de critères (une quantité plus importante de critères que l'autre), alors cette situation correspond bien à la notion intuitive d'incomparabilité. Dans cette situation, on ne peut pas dire si un élément est

préféré à un autre et l'on ne pourra donc pas mettre en place un préordre complet entre les différents éléments.

II.1.B. Éléments taxinomiques

Les méthodes d'aide à la décision multicritère ont comme objectif de proposer des moyens de déterminer un ordre de préférence (total ou partiel) ou classement d'un ensemble d'actions potentielles ou d'objets disponibles pour un problème, une situation donnée. Cependant, la nature de cet ordre dépend du problème traité. Les quatre problématiques de référence introduites par [ROY 85] et reprises par [MAY et al. 94] sont décrites comme suit (Tableau 2):

Problématique	Objectif	Résultat	Procédure
α	Choix d'un sous-ensemble contenant les actions «les meilleures» ou, à défaut, «satisfaisantes»	Choix	Sélection
β	Tri par affectation des actions à des catégories prédéfinies	Tri	Affectation
γ	Rangement de classes d'équivalence, composées d'actions, ces classes étant ordonnées de façon complète ou partielle	Rangement	Classement
δ	Description, dans un langage approprié, des actions et de leurs conséquences	Description	Cognition

Tableau 2. *Taxinomie des problématiques*

« On détermine le classement à l'aide de l'ensemble des critères de sélection, C . Pour chaque action potentielle a_i et chaque critère de sélection C_j , une valeur d'utilité en fonction d'un critère g_j , $g_j(a_i)$ (lire : « évaluation de l'action a_i selon le critère g_j ») est calculée par la fonction d'utilité du critère (fonction permettant d'attribuer une valeur d'importance au critère). L'ensemble des évaluations peut être représenté par un tableau à double entrée, appelé la matrice de performances, dans laquelle chaque ligne représente une action et chaque colonne un critère. » [FRA 96]

Voici un exemple de matrice de performances (Tableau 3) dans le cadre d'un achat/vente de produits boursiers.

Actions/Critères	Prix important	Prix faible	Expert vend	Expert achète
Acheter	faible	important	faible	important
Vendre	important	faible	important	faible

Tableau 3. *Matrice de performance*

« La plupart des méthodes multicritère utilisées en pratique demandent au décideur d'affecter une importance aux critères de sélection. Cette importance est traduite par un poids p_i , qui peut être soit ordinal, soit cardinal. Un poids ordinal définit un ordre d'importance entre les critères, où un critère peut être plus/moins/aussi important qu'un autre. Un poids cardinal est un chiffre, souvent un pourcentage, qui permet de mieux détailler les importances des critères, mais qui est plus difficile à exprimer par le décideur. » [FRA 96]

Il est souvent difficile pour les utilisateurs de définir leurs préférences directement sous forme de pondération, c'est pourquoi ces préférences sont le plus souvent déduites. L'utilisation de score à la place de pondération permet d'obtenir des écarts plus importants entre les critères composant le profil de l'utilisateur, et possède des avantages comme de ne pas nécessiter de normaliser les pondérations.

« Le concept d'importance des critères de sélection est, en réalité, présent dans tous les MCDA, même si elle n'est pas toujours exprimée par l'intermédiaire d'un poids. L'ordre des actions peut même être directement calculé par les valeurs d'utilité à travers le critère g_j , $g_j(a_i)$. Les poids et les valeurs d'utilité sont les données d'un processus d'agrégation, qui permet d'obtenir une valeur de préférence ou note globale pour chaque action. La valeur de préférence peut être une valeur absolue ou une valeur relative par rapport à une ou plusieurs autres actions, ce qui permet ensuite de déterminer le classement. » [FRA 96]

Pour ces processus d'agrégation, plusieurs méthodes ont été mises au point. Ainsi, l'on peut trouver par exemple, la somme pondérée et les opérateurs de moyenne ainsi que le minimum et maximum pondéré [DUB 86], la somme pondérée ordonnée (OWA) [YAG 88] ou encore les intégrales/mesures floues [SUG 74]. Ces méthodes sont décrites aux chapitres III.2.B et III.3.B.

« Les concepts de valeur de préférence et de note sont souvent appelés valeur d'utilité dans la littérature. Ces concepts sont appropriés pour des modèles de préférence qui utilisent une note unique par action, ce qui n'est pas le cas pour des relations valuées (on met une note sur une relation et non pas sur une action), par exemple. » [FRA 96]

Or dans la plupart des systèmes de recommandation, ce sont des objets, produits, ou actions qui sont valués. C'est pourquoi on utilise souvent la notion de valeur de préférence sous forme de note ou de score, ces valeurs représentant à quel point l'élément évalué est intéressant pour l'utilisateur.

L'identification du décideur (processus qui consiste à déterminer qui est le décideur) n'est néanmoins pas toujours simple. Un décideur n'est qu'un acteur parmi d'autres dans le processus de décision, qui est normalement obligé de prendre en compte l'avis d'autres acteurs pour aboutir à une solution.

Dans [ROY 85], un acteur est défini comme suit :

Un individu ou un groupe d'individus est acteur d'un processus de décision si, par son système de valeurs, que ce soit au premier degré du fait des intentions de cet individu ou groupe d'individus ou au second degré par la manière dont il fait intervenir ceux d'autres individus, il influence directement ou indirectement la décision. De plus, pour qu'un groupe d'individus (corps constitué ou collectivité) soit identifié comme un seul et même acteur, il faut que, relativement au processus, les systèmes de valeurs, systèmes informationnels et réseaux relationnels des divers membres du groupe n'aient pas à être différenciés.

Cette définition a été davantage détaillée par [JAC 81] et [MAY et al. 94]. Dans les problèmes les plus simples, le décideur est une personne pour qui il est suffisant de prendre en compte uniquement ses propres préférences pour prendre la décision. Il suffit ainsi d'exploiter uniquement les informations dans le profil de l'utilisateur pour l'assister. Un tel décideur a rarement vraiment besoin d'un support, sauf dans des cas où le nombre d'actions potentielles, d'objets disponibles devient trop grand.

« Au cas où ce décideur décide malgré tout d'utiliser un système d'aide à la décision, le problème principal devient le transfert de ses préférences vers le SAD retenu. Quel que soit ce SAD, il impose une certaine représentation des préférences du décideur. » [FRA 96]

Il est généralement admis que l'expression a priori de préférences du décideur est très difficile, ce qui explique l'apparition de méthodes permettant une obtention progressive des préférences. Ce sont les Méthodes Interactives d'Aide à la Décision, discutées dans la partie suivante.

II.1.C. Méthode Interactive d'Aide à la Décision

Vincke [VIN 89], définit une méthode interactive d'aide à la décision multicritère comme suit :

Définition : Une méthode interactive consiste en une alternance d'étapes de calculs et d'étapes de dialogue avec le décideur. La première étape de calculs fournit une première solution. Celle-ci est présentée au décideur qui réagit en apportant des informations supplémentaires sur ses préférences (étape de dialogue). Cette information est injectée dans le modèle utilisé et permet de construire une nouvelle solution.

Le dialogue signifie dans ce cas une contribution active du décideur à la construction de la solution au lieu de définir a priori les paramètres du calcul utilisé, comme c'est le cas dans les systèmes d'analyse de décision multicritère (MCDA).

Pour ce faire, le processus utilisé est un processus interactif. Dans lequel le système associé présente à chaque étape ses recommandations au décideur. Celui-ci estime si ce résultat qui lui est présenté est satisfaisant ou non à partir des informations fournies. Dans le cas où il n'est pas satisfait, il peut continuer à introduire de l'information sur ses préférences, en continuant à utiliser le système ou en fournissant des notes sur des éléments proposés afin trouver des solutions plus satisfaisantes. Cette procédure peut être illustrée par le schéma suivant (Figure 4) extrait des travaux de [DON 92].

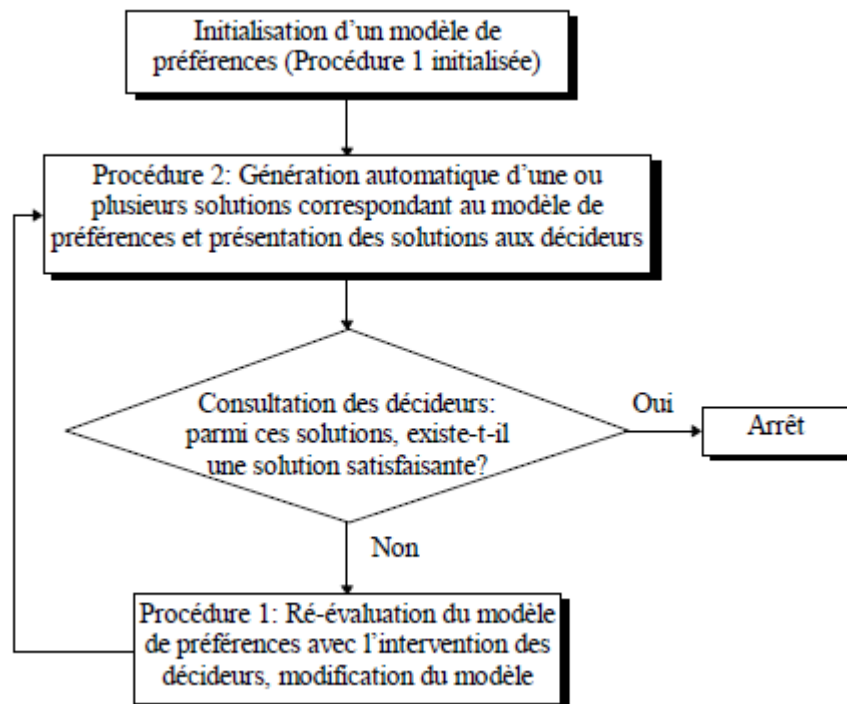


Figure 4. *Processus interactif de décision*

L'objectif d'une telle méthodologie interactive n'est pas d'identifier un modèle de préférence préexistant dans la tête du décideur, mais de le faire participer en ajoutant des informations dans le système jusqu'à ce qu'il soit satisfait.

« L'interactivité permet également au décideur de mieux comprendre ses préférences concernant le problème traité et de les modifier en fonction d'une compréhension progressivement améliorée. Cette approche facilite en même temps l'acceptation des recommandations du système, puisque le décideur peut continuer le dialogue jusqu'à ce qu'il soit convaincu que le modèle de préférence est bien le sien. » [FRA 96]

Ce fonctionnement permet de mettre en évidence que les utilisateurs ne sont pas, la plupart du temps, objectifs vis-à-vis de leurs préférences. Cette méthodologie leur permet de corriger la vision qu'ils ont par rapport à ce qui est important pour eux. Ce fonctionnement implique que le système s'améliore à chaque information reçue et qu'il n'y a pas de problèmes de bruit des informations.

« Les modifications dans les préférences du décideur font partie de la rationalité limitée générale des décideurs, qui est un aspect bien évoqué dans [SIM 55] et davantage approfondi dans [SIM 83]. Pour cette raison, il est impossible ou non souhaitable d'essayer

de déterminer automatiquement si une recommandation est vraiment satisfaisante pour le décideur ou non. Simon constate que seul le décideur lui-même peut déterminer son niveau de satisfaction concernant une recommandation (satisfying). C'est pour cette raison qu'une approche d'optimisation classique n'est pas appropriée pour résoudre les problèmes de décision multicritère. » [FRA 96]

L'utilisation du principe d'interactivité permet ainsi à l'utilisateur de déterminer de lui-même quand il est satisfait des éléments qui lui sont proposés. La rationalité limitée des utilisateurs nécessite d'être prise en compte à travers des systèmes permettant à l'utilisateur de fournir des informations contradictoires. C'est le cas lorsqu'il change de préférences, il faut donc modéliser les préférences de l'utilisateur et avoir la capacité de mettre à jour ce modèle rapidement, et cela sans être bloqué par les informations précédemment fournies (comme le serait un système à base de règles). C'est-à-dire qu'il faut prendre en compte les informations précédemment obtenues sur l'utilisateur, mais en cas de conflit, il faut utiliser les informations les plus récentes sans pour autant ne pas tenir compte des anciennes. Il faut ainsi privilégier les nouvelles informations en utilisant par exemple une distribution de gradient.

Un regroupement peut être fait entre le domaine de l'aide à la décision et les méthodes d'agrégation. Ce regroupement est appelé Procédures d'Agrégation Multicritère (PAMC). Plus d'informations peuvent être trouvées dans [HAM et al. 07].

Afin de modéliser numériquement les préférences d'un utilisateur, la théorie multiattribut a été développée.

II.2. Théorie Multi-Attributs

« Les premières tentatives d'aide à la décision avec objectifs multiples ou non remontent à la fin des années 1960 à travers les travaux de Raïffa et Edwards [EDW 71, RAI 69] qui ont donné naissance à l'analyse de la décision (Decision Analysis). Dans ces travaux, le modèle formel de référence fait appel à la représentation numérique des préférences du décideur sur l'ensemble des choix possibles grâce à une fonction de préférence dite fonction d'utilité. L'idée de base d'une telle démarche étant que l'encodage (c'est-à-dire la construction) d'une fonction d'utilité dans un contexte de décision donné permettra d'affecter des "scores" ou utilités, aux actions potentielles (c'est-à-dire les choix possibles) auxquelles fait face le décideur. Ces scores permettront ensuite de classer les actions de la moins désirable à la plus désirable (et vice-versa). » [BOUY et al. 06]

On parle de fonction d'utilité multiattribut vis-à-vis d'une "combinaison simple" de fonctions d'utilité mono-objectif. Ainsi l'objectif de la théorie de l'utilité multiattribut est de modéliser numériquement les préférences d'un décideur.

Le concept d'utilité est relativement simple : il s'agit d'utiliser une fonction $U : X \rightarrow \mathbb{R}$ attachant à tout objet de l'ensemble de choix un nombre réel de telle sorte que plus ce dernier est élevé, plus l'objet est "préféré" par le décideur.

Soit $x := (x_1 \times \dots \times x_n)$, avec $n \geq 2$, un ensemble d'objets d'intérêt décrits par un ensemble $\mathbb{N} := \{1, \dots, n\}$ d'attributs.

Ces préférences prennent mathématiquement la forme d'une relation binaire \succsim qu'il s'agit de représenter par le biais d'une fonction d'utilité globale telle que :

$$x \succsim y \Leftrightarrow U(x) \geq U(y), \quad \forall x, y \in X \quad (1)$$

Pour la fonction d'utilité globale, nous considérons le modèle transitif décomposable de Krantz, Luce, Suppes & Tversky [KRA et al. 71]) (voir aussi [BOUY 04a] et [BOUY 04b]), dans lequel U est défini par :

$$U(x) := F(u_1(x_1), \dots, u_n(x_n)), \quad \forall x = (x_1, \dots, x_n) \in X \quad (2)$$

Où les fonctions $U_i : X_i \rightarrow \mathbb{R}$ sont appelées les fonctions d'utilité, et $F : \mathbb{R}^n \rightarrow \mathbb{R}$, non décroissantes en chacun de ses arguments, est parfois appelée la fonction d'agrégation.

Un des principaux avantages des systèmes multiattribut est qu'il n'y a pas de notion d'incomparabilité d'un objet par rapport à un autre. En effet, cette approche permet, contrairement aux méthodes multicritère, de toujours comparer des objets entre eux. Et cela est possible grâce au score calculé pour chaque objet, reflétant son intérêt, ainsi on obtient un pré-ordre total, et ce quel que soit les objets rencontrés et leurs caractéristiques intrinsèques et extrinsèques. De plus, la théorie de l'utilité multiattribut (MAUT) offre la possibilité d'intégrer à la fois des critères quantitatifs et qualitatifs, ainsi que d'aboutir à une hiérarchisation stricte des solutions en présence. Elle permet donc de réduire l'incertitude liée au choix d'implantation (information incomplète sur les ressources génériques et spécifiques aux différents sites) et à la rationalité du décideur (préférences subjectives).

II.3. Conclusion

Ainsi dans le cadre de l'aide à la décision, des techniques multicritère ou multiattribut peuvent être utilisées. Le choix entre les 2 dépend principalement du type de données que l'on utilise, des critères ou des attributs ?

En dehors de cette considération, la méthode multiattribut offre des avantages importants tels que l'absence de problème d'incomparabilité et permet la construction d'ordre total entre des éléments.

Nous avons détaillé la notion de système de recommandation et la mise en œuvre de techniques multicritère/multiattribut. Il nous faut à présent détailler les techniques d'apprentissages de profils/préférences, nécessaires pour faire évoluer le profil de l'utilisateur.

Chapitre III. Prise en compte et évolution de profils/préférences

On cherche à avoir un ou plusieurs profils représentant au mieux les préférences d'un utilisateur. En effet, ce sont ces profils qui vont permettre d'effectuer une recommandation personnalisée d'objets adaptés à chaque utilisateur. Pour cela, on utilise des techniques (souvent issue de l'intelligence artificielle) afin d'effectuer l'apprentissage de profils/préférences des utilisateurs. Cet apprentissage passe par plusieurs étapes. En effet, il faut décomposer les objets sélectionnés ou évalués par l'utilisateur afin de déterminer ce qui est important pour lui. Il faut ensuite mettre à jour le profil multicritère de l'utilisateur avec les nouvelles informations obtenues et enfin utiliser les données de son profil afin de sélectionner et trier (filtrer) les objets proposés à l'utilisateur (Figure 5).

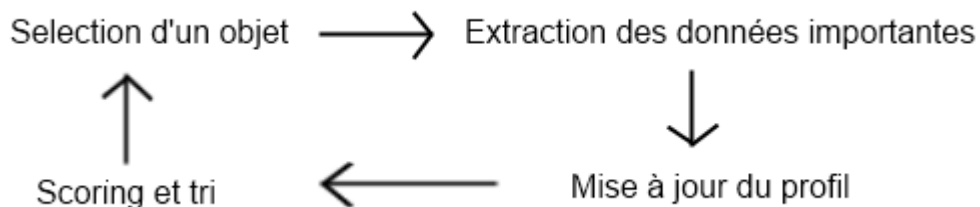


Figure 5. Cycle d'apprentissage de préférences

III.1. Apprentissage de profils/préférences

L'apprentissage de profils/préférences (ou *preference learning*) des utilisateurs permet, en exploitant ces profils/préférences, de produire des suggestions, des recommandations aux utilisateurs. Plus d'informations peuvent être trouvées dans [FUR 10].

Avant de définir les méthodes employées pour réaliser cet apprentissage et afin de bien comprendre l'intérêt d'apprendre les profils/préférences des utilisateurs, nous allons expliquer le cadre de leur utilisation.

III.1.A. Exploitation des profils/préférences

Pour exploiter des profils/préférences utilisateurs, afin d'effectuer des traitements sur les solutions présentées à l'utilisateur pour représenter ses préférences, il existe actuellement deux grands principes. La classification, et le ranking.

III.1.A.i. La classification

La classification consiste à assigner un groupe à chaque objet, en fonction de ses caractéristiques. On ne cherche pas ici à créer un ordre entre les objets, mais à les rassembler en fonction de caractéristiques définies pour chaque groupe.

Avec un ensemble d'objets $\{x_k \mid k = 1 \dots n\} \subseteq \mathcal{X}$ et un ensemble de groupe $\{\lambda_i \mid i = 1 \dots c\}$. Un groupe λ_i est assigné à chaque objet x_k . Cela implique la préférence suivante : $\lambda_i \succ_{x_k} \lambda_j$. Cela signifie que l'objet x_k possède des caractéristiques plus proches de celles que l'on peut trouver au sein du groupe λ_i plutôt qu'au sein du groupe λ_j .

Ainsi, cela permet de regrouper les solutions proposées à l'utilisateur en fonctions de caractéristiques propres aux groupes. Et donc, en fonction de ce que recherche l'utilisateur, en fonction de ses préférences, on constitue des groupes de solutions.

Par contre, cette méthode ne permet pas de définir d'ordre total entre les objets, ce qui peut être problématique, puisque dans ce cas-là, nous n'avons pas la possibilité de sélectionner et de trier les éléments qui vont le plus satisfaire l'utilisateur.

III.1.A.ii. Le ranking

Le ranking consiste à déterminer un ordre entre les différentes solutions, dans le but de pouvoir sélectionner et trier les éléments qui vont le plus satisfaire l'utilisateur.

Le ranking [LIU 09] est une technique d'apprentissage supervisée ou semi-supervisée pour des problèmes dans lesquels le but est de construire un ordre total (classement des objets entre eux, à différencier de la classification présentée précédemment) à partir d'une liste d'objets. Cet ordre est typiquement mis en place par l'introduction d'un système de note ou score (par exemple une échelle de valeurs de 0 à 5 pour chaque objet). Le modèle de ranking propose de faire une permutation entre les objets qui n'ont pas encore été appris. Ces permutations sont effectuées en fonction du score global obtenu pour chaque élément.

Afin de pouvoir définir le ranking, il nous faut définir la notion de transitivité :

Définition de la transitivité :

Une relation binaire \mathcal{R} définie sur un ensemble E est transitive quand, à chaque fois que l'on a trois éléments x , y et z appartenant à E , et que x et y sont en relation, ainsi qu' y et z , alors x et z sont en relation.

Exemple : Si $A > B$, et $B > C$, alors $A > C$.

Définition du ranking :

Avec un ensemble d'objets $\{x_k \mid k = 1 \dots n\} \subseteq \mathcal{X}$, chaque objet x_k est associé à un ordre total vis-à-vis d'autres éléments, et où l'ordre entre les exemples est transitif.

Parmi les principales méthodes pour réaliser de l'apprentissage de préférences, l'on peut trouver l'utilisation de réseaux de neurones et l'apprentissage symbolique automatique (par exemple par un système d'apprentissage par renforcement) que nous définirons par la suite. Mais également d'autres méthodes que nous ne définirons pas ici, notamment les algorithmes génétiques (en anglais : genetic algorithms) [HOLL 75], [DE JONG 75], le raisonnement à base de cas (en anglais : case-based learning) [KOL 87], ainsi que des réseaux Bayésiens [DAN 07].

On commencera par présenter les réseaux de neurones qui sont très utilisés pour effectuer l'apprentissage de profils/préférences.

III.1.B.Les réseaux neuronaux

Les réseaux neuronaux [BIS 06] ont été utilisés dans de nombreux domaines d'application : mémoires associatives, diagnostic, traitement du signal, reconnaissance d'images, reconnaissance de la parole, compression de données, contrôle, prévision, analyse de données, etc. Un réseau neuronal est essentiellement un outil mathématique permettant d'identifier des fonctions mathématiques. L'utilisation d'un réseau de neurones peut être considérée comme la mise à jour des poids des connexions au sein du réseau, afin de réussir la tâche qui lui est demandée.

Leur conception très schématiquement (FIG 6) est inspirée du fonctionnement de vrais neurones (humains ou non).

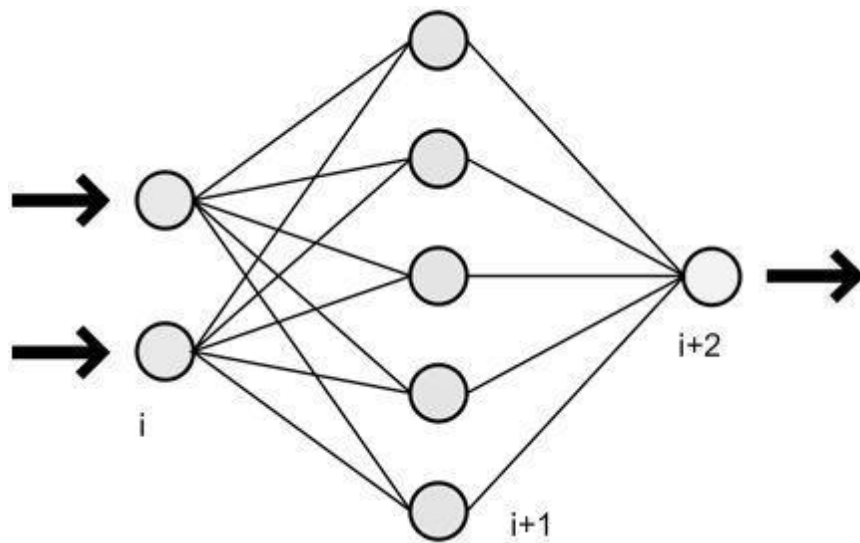


Figure 6. *Exemple de réseau de neurones*

Un réseau de neurones peut être, par exemple, composé d'une succession de couches dont chacune prend ses entrées sur les sorties de la précédente (voir FIG 6). Chaque couche i est composée de N_i neurones, prenant leurs entrées sur les $N_i - 1$ neurones de la couche précédente.

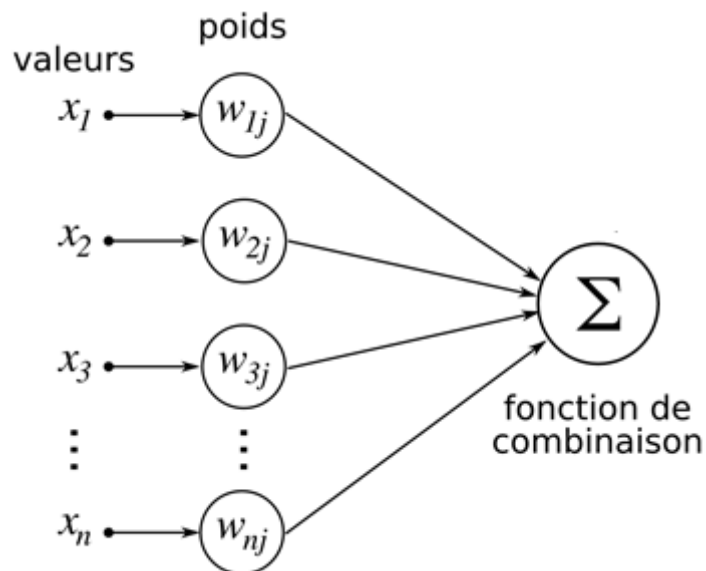


Figure 7. *Exemple de nœud*

À chaque nœud est associé un poids synaptique (w_{1j}), de sorte que les $N_i - 1$ sont multipliés par ce poids (Figure 7, fonction de combinaison), puis additionnés par les neurones de niveau i , ce qui est équivalent à multiplier le vecteur d'entrée par une matrice de transformation. Le fonctionnement des neurones est identique aux perceptrons, à la différence près qu'ils utilisent une fonction sigmoïde à la place de la fonction signe.

Dans le cas qui nous intéresse, les réseaux neuronaux sont utilisés pour l'identification des fonctions de préférence des décideurs (on effectue un apprentissage de leurs préférences). Une fois identifiées, on peut alors les utiliser pour évaluer des solutions présentées à l'utilisateur.

Pour illustrer le principe de l'apprentissage via un réseau de neurones plus en détail, nous prendrons l'exemple du système INKA développé par Främling [FRA 96].

« Il y a déjà eu certains travaux sur l'utilisation des réseaux de neurones comme méthodes d'analyse multicritère. Des réseaux neuronaux multicouches à rétropropagation étaient utilisés dans [WAN 92] pour apprendre la fonction de préférence des décideurs à partir d'exemples. Les exemples utilisés dans [WAN 92] se limitent à des problèmes à deux critères de sélection. Ils se limitent également à des fonctions de préférence monotones, ce qui n'est pas toujours le cas des fonctions de préférence réelles. De plus, il s'agit dans leur cas notamment d'un Système d'Aide à la Décision (SAD) non interactif. » [FRA 96]

Or, dans la plupart des cas, le nombre de critères à prendre en compte pour effectuer la prise de décision dépasse 2. On remarquera que très souvent dans la littérature, les cas de prise de décisions sont modélisés avec une quantité réduite de critères pris en compte. Ce qui n'est pas représentatif des conditions réelles.

« L'identification d'une fonction de préférence univoque suppose que le décideur soit rationnel et cohérent, même si le décideur humain ne peut probablement jamais l'être à 100 %. Il est difficile de représenter un décideur qui un jour donne une réponse et le lendemain son contraire ou une alternative très différente. En effet, les décideurs ne sont pas toujours rationnels, c'est ce qu'on appelle la rationalité limitée. En d'autres termes, l'acteur est rationnel s'il préfère A à B et B à C, on peut en déduire qu'il préfère A à C (transitivité). Cependant, lors d'un choix complexe, il cherche moins à étudier l'ensemble des possibilités qu'à trouver une solution raisonnable dans une situation d'incertitude. Il va s'arrêter généralement à la première option qui satisfera à la situation concrète, tout en évitant de consommer trop de temps à effectuer son choix. Dans le cas d'un tel décideur, un réseau neuronal qui fonctionne par des principes de régression ne donnera ni la première réponse ni la deuxième (son contraire), mais la moyenne des deux. » [FRA 96]

« L'apprentissage automatique des réseaux de neurones s'effectue traditionnellement uniquement par modification des poids du réseau. Par contre, le nombre de neurones est défini par l'utilisateur avant l'apprentissage et reste le même jusqu'à la fin de l'apprentissage. Il existe néanmoins un certain nombre d'algorithmes d'apprentissage qui apprennent par ajout et suppression de neurones [JOK 91], [FRA et al. 95]. Certains algorithmes combinent les deux approches, comme ceux qui sont décrits dans [PLA 91] et [DECO 93].

Il existe un grand nombre de règles d'apprentissage, dont celles de Hebb [HEB 49], du perceptron [ROS 58] et de Widrow (ou de Widrow/Hoff) [WID 60], qui ont d'ailleurs servi de base pour la plupart des règles de modification de poids développées par la suite. La règle d'apprentissage de Widrow est actuellement celle qui est la plus utilisée, elle est notamment utilisée dans le réseau de rétropropagation. Ceci a provoqué le développement d'un grand nombre de versions améliorées de la règle d'origine, qui ont contribué à la notoriété des réseaux de rétropropagation. » [FRA 96]

Un inconvénient important se trouve dans la nécessité qu'ont les réseaux de neurones à respecter la notion de transitivité. Un autre inconvénient est que ce modèle ne prend pas en compte la notion d'incomparabilité (présentée en II.1.A.iv). De plus, un réseau de neurones ne permet pas à l'utilisateur de changer de préférences de manière importante, et nécessite une base d'apprentissage afin de pouvoir être efficace.

Un autre système d'apprentissage de profils/préférences est défini en tant qu'apprentissage symbolique automatique, et plus précisément, il s'agit de l'apprentissage par renforcement appliqué aux profils utilisateurs. On peut remarquer que celui-ci ne nécessite pas de base d'apprentissage.

III.1.C. Apprentissage par renforcement pour les profils utilisateurs

La méthode d'apprentissage par renforcement appliquée aux profils utilisateurs a surtout été présentée et expérimentée dans le cadre des travaux de recherches suivants [BOUG et al. 04], [TMA 02]. Cette méthode a été utilisée dans un système qui effectuait en partie de la Recherche d'Information (RI) mais surtout du filtrage d'information. Cette méthode est décrite comme suit :

L'apprentissage des profils que nous utilisons est basé sur un principe de renforcement [SUT 98]. À cet effet, nous considérons $po_{ik}^{(t)}$ correspondant au poids du critère k de l'objet $o_i^{(t)}$ est sélectionné et jugé pertinent, il faut trouver une représentation du profil $p_x^{(t)}$ au même temps t qui permet de retrouver cet objet avec un score " fort ", soit λ , puis intégrer ces données pour apprendre le profil global de l'utilisateur. Ceci revient donc à chercher les $pc_j^{(t)}$, poids de l'attribut a_j dans le profil $p_x^{(t)}$, qui satisfont l'équation suivante :

$$\sum_{k \in o_i^{(t)}, a_j \in p_x^{(t)}} po_{ik}^{(t)} \cdot pc_j^{(t)} = \lambda \quad (3)$$

Pour l'intégration dans le profil de l'utilisateur, il est utilisé la formule de distribution du gradient suivante :

$$p_i^{(t+1)} = p_i^{(t)} + 0.1 * \log(1 + pc_i^{(t)}) \quad (4)$$

- $p_i^{(t+1)}$ correspond au poids du critère i au temps $t+1$
- $p_i^{(t)}$ correspond au poids du critère i au temps t
- $pc_i^{(t)}$ correspond au poids calculé pour le critère i à partir de l'objet courant

C'est sur cette méthode que se basent nos travaux. Évidemment, il y a de nombreuses limitations pour l'utilisation des données introduites par cette méthode et en particulier la méthode employée pour déterminer les poids intermédiaires. En effet, le système proposé par Boughanem et al. [BOUG et al. 04] a pour but un système de filtrage et n'est utilisé que pour des données textuelles. Ainsi toute la partie détaillant la méthode d'obtention des poids intermédiaires, ne peut être utilisée, car les méthodes appliquées ne sont utilisables que pour des données de types textes. Or dans notre système, les types de données utilisées sont divers, et varient en fonction des objets et de leur description. De plus, à partir des données mises en œuvre, il n'existe pas de méthodes permettant de se ramener à des données de type texte uniquement.

L'un des enjeux principaux dans les systèmes d'apprentissage de profils utilisateurs consiste en une décomposition des données fournies par l'utilisateur, autrement appelée désagrégation.

III.2. Désagrégation et agrégation avec indépendance des critères

Il est nécessaire de déterminer quels attributs sont importants ou non pour l'utilisateur dans un objet donné et pour se faire il faut décomposer cet objet. Ces attributs sont considérés comme indépendants les uns des autres, car dans la majorité des cas l'on ne connaît pas quels sont les interactions, les dépendances, les liens d'un critère par rapport à un autre. Mais cela n'est pas satisfaisant, car dans un cadre d'utilisation réelle, les attributs sont liés entre eux, ils présentent des interactions les uns avec les autres. Par exemple, pour une voiture, sa puissance est liée à son moteur. Si ces attributs sont considérés comme indépendants, on perd de l'information qui pourrait être utile. Donc on sera moins précis dans nos recommandations en ne prenant pas en compte ces informations, et ainsi nos recommandations seront de moindres qualités. Cependant, il est très difficile d'avoir premièrement, un système qui puisse gérer ces interactions entre les attributs, et deuxièmement, de connaître ces interactions, ces liens. C'est pourquoi, dans un premier temps, les critères sont considérés comme étant indépendants les uns avec les autres.

III.2.A. Opérateurs de désagrégation

De manière générale, on ne cherche pas à construire directement une fonction d'utilité afin de représenter les préférences du décideur, mais on essaye plutôt d'en construire une, d'une forme particulière dont on sait que sa construction se révélera « réalisable » cognitivement.

De multiples formes ont été décrites dans la littérature, dont voici certainement ci-dessous les principales. Dans cette liste, on note X_i l'ensemble des valeurs que peut prendre le $i^{ème}$ attribut et l'on suppose que $X \subseteq \prod_{i=1}^n X_i$. Les axiomatiques assurant l'existence de ces formes diffèrent suivant que l'on se place dans le choix dans le certain ou bien dans le contexte de l'espérance d'utilité (EU), c'est-à-dire lorsque $U(\cdot) = \sum_j p_j u(x^j)$ (comme dans Von Neumann et Morgenstern [NEU 44]), le contexte d'application est indiqué pour chaque référence de la liste. Voici une description succincte de plusieurs d'entre elles issues de [BOUY et al. 06].

III.2.A.i. La décomposition additive

Il existe des fonctions $u_i : X_i \mapsto \mathbb{R}$ telles que

$$(x_1, \dots, x_n) = \sum_{i=1}^n u_i(x_i) \quad (5)$$

Parmi les ouvrages de référence pour cette décomposition figure [FIS 70, chapitres 4 et 5], [KRA et al. 71, chapitre 6], [KEE 93, chapitre 3], [LUC 64], [DEB 60] et [WAK 89, chapitre 3] pour le certain, et [FIS 70, chapitre 11] et [KEE 93, chapitre 5 et 6] dans le cadre de l'espérance d'utilité.

III.2.A.ii. La décomposition multiplicative

Il existe des fonctions $u_i : X_i \mapsto \mathbb{R}$ telles que

$$u(x_1, \dots, x_n) = \prod_{i=1}^n u_i(x_i) \quad (6)$$

Cette décomposition est proche de la précédente et s'y ramène en utilisant une échelle logarithmique dans le cas des $u_i \geq 0$.

III.2.A.iii. La décomposition multilinéaire

Cette méthode est également appelée décomposition polynomiale ou encore multiplicative-additive. Il existe des fonctions $u_i : X_i \mapsto \mathbb{R}$ et pour tout $j \in J$, ensemble des parties de $\{1, \dots, n\}$, il existe $\pi_j \in \mathbb{R}$ tels que

$$u(x_1, \dots, x_n) = \sum_{j \in J} \pi_j \prod_{k \in j} u_k(x_k) \quad (7)$$

Cette décomposition est présentée dans [KRA et al. 71, chapitre 7], [FIS 75] et [FUH 91] pour le certain, et dans [KEE 93, chapitre 5 et 6] et [FARQ 81] pour l'espérance d'utilité.

III.2.A.iv. La structure décomposable

Il existe des fonctions $u_i : X_i \mapsto \mathbb{R}$ et une fonction $F : \mathbb{R}^n \mapsto \mathbb{R}$ telles que

$$u(x_1, \dots, x_n) = F(u_1(x_1), \dots, u_n(x_n)) \quad (8)$$

Bouyssou et Pirlot [BOUY 02] et Krantz et al. [KRA et al. 71, chapitre 7] ont étudié cette représentation dans le certain. Cette structure est plus générale que les précédentes, mais elle présente un inconvénient majeur : l'unicité des u_i et de F n'est pas assurée, ce qui peut poser des problèmes de construction.

III.2.A.v. La décomposition additive non transitive

Il existe des fonctions $v_i : X_i \times X_i \mapsto \mathbb{R}$ telles que

$$x \succeq y \Leftrightarrow \sum_{i=1}^n v_i(x_i, y_i) \geq 0 \quad (9)$$

Voir [FIS 91] ou [BOUY 02] dans le certain et [NAK 90] pour une généralisation du critère de l'espérance d'utilité.

Parmi les fonctions additives non transitives se trouve le cas particulier de la décomposition additive par différence : il existe des fonctions $u_i : X_i \mapsto \mathbb{R}$ et des fonctions $F_i : \mathbb{R} \mapsto \mathbb{R}$ telles que

$$x \succeq y \Leftrightarrow \sum_{i=1}^n F_i(u_i(x_i) - u_i(y_i)) \geq 0 \quad (10)$$

Voir [TVE 69] et [BOUY 02] dans le certain.

L'opération inverse correspond à rassembler des diverses données afin de déterminer une valeur correspondant à l'intérêt qu'à l'utilisateur pour l'élément. Cette opération est appelée agrégation.

III.2.B. Opérateurs d'agrégation

Dans le but d'obtenir une valeur représentant l'importance d'un objet pour l'utilisateur, il est nécessaire d'avoir des opérateurs qui agrègent les scores de chaque attribut afin de pouvoir obtenir un score global pour l'objet.

Voici une description succincte de plusieurs d'entre elles issues de [GRA 02].

III.2.B.i. *La somme pondérée ou moyenne arithmétique pondérée*

Elle est définie par

$$\psi(a_1, \dots, a_n) = \sum_{i=1}^n w_i a_i \quad (11)$$

où les $w_i \in [0, 1]$ sont des poids, tels que

$$\sum_{i=1}^n w_i = 1 \quad (12)$$

D'autres types de moyenne existent (géométriques, harmoniques, *etc.*), qui peuvent toutes s'écrire sous la forme :

$$M_f(a_1, \dots, a_n) = f^{-1}\left(\sum_{i=1}^n w_i f(a_i)\right) \quad (13)$$

Voir [DUB 86].

Où f est une fonction continue strictement croissante (moyennes généralisées). Toutes les moyennes généralisées sont idempotentes, continues, strictement monotones. Seule la somme pondérée vérifie la stabilité au changement d'échelle linéaire.

III.2.B.ii. Le minimum et maximum pondérés

Ils ont été introduits par Dubois et Prade dans le cadre de la théorie des possibilités [DUB 86]. Soit $w = (w_1, \dots, w_n)$ un vecteur de poids, $w_i \in [0, 1]$, tel que

$$\bigvee_{i=1}^n w_i = 1 \quad (14)$$

Ils sont définis par :

$$wmin_w(a_1, \dots, a_n) = \bigwedge_{i=1}^n [(1 - w_i) \vee a_i] \quad (15)$$

Voir [DUB et al. 98].

Et

$$wmax_w(a_1, \dots, a_n) = \bigvee_{i=1}^n [w_i \wedge a_i] \quad (16)$$

Voir [FAR et al. 99]. Le minimum et maximum usuel sont obtenus par $w = (1, 1, \dots, 1)$. Ils vérifient les propriétés d'idempotences, de continuité et de monotonie.

III.2.B.iii. La somme pondérée ordonnée (OWA)

L'opérateur OWA (Ordered Weighted Average) a été introduit par Yager [YAG 88]. Il est défini par :

$$OWA_w(a_1, \dots, a_n) = \sum_{i=1}^n w_i a_{(i)} \quad (17)$$

Avec $w = (w_1, \dots, w_n)$ un vecteur de poids, $w_i \in [0, 1]$ tel que $\sum_{i=1}^n w_i = 1$, et où la notation (\cdot) indique une permutation des indices telle que $a_{(1)} \leq \dots \leq a_{(n)}$. Ainsi, le poids n'est pas sur les sources, mais sur le rang des quantités.

Ainsi nous avons à disposition de nombreuses méthodes permettant de désagréger ou d'agréger des données. Cependant, il faut définir comment utiliser ces données afin de comparer des éléments entre eux.

III.2.C. Comparaison d'éléments

Il est nécessaire pour effectuer un ranking des éléments présentés à l'utilisateur, de définir des méthodes permettant de les comparer.

Il y a deux manières naturelles de représenter des préférences vis-à-vis d'un ensemble d'éléments, à savoir, d'évaluer les éléments de manière individuelle et de comparer les éléments entre eux [FUR 05].

- La première idée est d'estimer une sorte de degré d'utilité (score) pour chaque élément. Pour obtenir un classement, par exemple, les éléments peuvent être ordonnés en fonction de leur degré d'utilité respectif. Cette approche est dite « école Américaine ».
- La seconde approche propose d'utiliser un prédicat binaire, qui prédise si on a $\lambda_i \succ_{x_k} \lambda_j$ ou $\lambda_j \succ_{x_k} \lambda_i$ pour un exemple x . Et pour obtenir le classement, on compare les éléments 2 à 2 (par paire). Cette approche est dite « école Européenne ».

Dans cette seconde approche, afin de déterminer la meilleure alternative pour une paire d'éléments, le concept de Pareto optimalité est souvent utilisé.

Ce concept est défini par : *une alternative a est dite Pareto optimale si elle n'est dominée par aucune autre. Une alternative qui domine au sens de Pareto ne peut être améliorée au regard d'un critère sans la détériorer pour un autre.*

Grâce à cela, on peut ordonner les éléments entre eux. Cependant, cette approche fait intervenir des problèmes d'incomparabilités. Ce qui ne permet pas de définir un préordre total (chaque élément est ordonné par rapport aux autres).

Voici un exemple :

Supposons que nous cherchions à comparer deux objets A et B caractérisés par deux critères, dont les vecteurs d'utilité sont $A \rightarrow (10 ; 5)$ et $B \rightarrow (8 ; 4)$. Il est « intuitivement évident » que le premier objet est « supérieur » au second puisque pour chaque attribut, il est supérieur au second ($10 > 8$), et ($5 > 4$).

En revanche, il paraît impossible de classer les objets $A \rightarrow (10 ; 5)$ et $C \rightarrow (6 ; 11)$. En effet, le premier attribut est préféré dans A , car ($10 > 6$), alors que le second est préféré dans C , car ($11 > 5$). Et dans ce cas, on ne voit pas pourquoi l'un devrait prévaloir sur l'autre.

La première méthode permet de calculer une valeur globale représentant l'intérêt de l'élément (objet), pour l'utilisateur. Cette valeur globale correspond à une agrégation de plusieurs scores représentant chacun l'intérêt (la performance) d'un critère pour l'utilisateur. Cette méthode offre la possibilité de comparer des éléments très différents. Par exemple, on peut avoir deux objets qui n'ont rien à voir entre eux, comme une voiture et une maison, et grâce à cette méthode il reste possible de déterminer lequel est préféré des deux pour l'utilisateur. Cela revient à dire que cette méthode ne souffre pas d'incomparabilité.

Nous n'avons utilisé jusqu'à présent que des critères que nous considérons comme indépendants les uns des autres. Or dans la majorité des cas, les critères sont dépendants les uns des autres. C'est pour cela que des méthodes intégrant les relations, les dépendances et l'interaction entre les critères ont été développées, et en particulier pour effectuer des décompositions et des agrégations.

III.3. Désagrégation/agrégation avec interaction entre les critères

Un des aspects significatifs dans les problèmes d'agrégation est la prise en considération de l'importance des attributs ou critères considérés, laquelle est habituellement modélisée par l'utilisation de poids ou de score associés aux attributs ou critères. Puisque ces poids doivent être pris en considération durant la phase d'agrégation, il est nécessaire d'utiliser des fonctions d'agrégations pondérées, abandonnant ainsi la propriété de symétrie. Jusqu'à récemment, les fonctions d'agrégations pondérées les plus utilisées étaient des fonctions de type moyennes, telles que les moyennes quasi linéaires.

« Cependant, les moyennes arithmétiques pondérées et, plus généralement, les moyennes quasi linéaires présentent certaines faiblesses. Aucune de ces fonctions n'est capable de modéliser une quelconque interaction parmi les attributs. En effet, il est bien connu en théorie de l'utilité multiattribut (MAUT) que ces fonctions conduisent à l'indépendance préférentielle mutuelle (voir par exemple [FIS 95]) parmi les attributs, qui, dans un certain sens, exprime l'indépendance des attributs. Comme ces fonctions ne sont pas appropriées en présence d'attributs dépendants, la tendance a été de construire des attributs censés être indépendants, ce qui entraînait souvent des erreurs dans les évaluations.

Dans le but d'obtenir une représentation flexible des phénomènes complexes d'interactions parmi les attributs ou critères (par exemple une synergie positive ou négative entre certains critères), il s'est avéré utile de remplacer le vecteur poids par une fonction d'ensemble non additive, permettant ainsi de définir un poids non seulement sur chaque critère, mais aussi sur chaque sous-ensemble de critères." [BOUY et al. 06]

La tendance à utiliser des critères indépendants les uns des autres mène à une perte d'informations. En effet, pour produire de bonne recommandation il est nécessaire d'avoir une bonne modélisation des préférences de l'utilisateur et cela nécessite de nombreuses informations (pas trop, sinon cela devient du bruit).

Ainsi, si l'on prend l'exemple d'un achat de voiture, un utilisateur voudra un moteur puissant et des freins puissants également. Si les critères sont considérés comme indépendants, nous ne possédons pas cette information à propos des préférences de l'utilisateur, et donc les recommandations effectuées seront de moindre qualité. De plus, l'utilisation des interactions entre les critères permet de déterminer les critères qui interagissent le plus et permet ainsi de déterminer les critères qui sont les plus utiles et de réduire le bruit dans les données. En effet, on peut ainsi détecter les critères qui ne sont jamais utilisés ou qui ne satisfont pas, et réduire leur impact lors de la recommandation.

Une première méthode d'agrégation prenant en compte les liaisons entre les critères consiste à utiliser un système expert ou des règles de production.

III.3.A. Méthode par système expert / règles de production

"Une méthode existe basée sur un apprentissage qui permet de trouver des liens de dépendance entre des symboles, souvent sous forme de règles de production. Cet apprentissage permet la construction de systèmes experts à partir d'exemples. Certaines méthodes sont également capables de traiter et de générer des probabilités ou coefficients de certitude, parmi lesquelles la méthode des arbres de décision probabilistes de Quinlan, ID3 [QUI, 86, 87a, 90] et C4 [QUI, 87b], qui est certainement la plus connue. D'autres méthodes sont présentées par exemple dans [KOD 90]." [FRA 96]

Mais cette méthode présente des limites, en effet, ce type de système est très difficile à maintenir. En effet lorsque l'utilisateur est incohérent, le système n'arrive plus à gérer les règles/probabilités introduites dans le système, car devenues incohérentes, et ainsi le système n'est plus fonctionnel.

C'est pourquoi, dans le but d'obtenir une représentation flexible des phénomènes complexes d'interactions parmi les attributs ou critères que l'utilisation des *mesures floues* a été proposée par Sugeno [SUG 74] pour généraliser les mesures additives.

III.3.B. Mesures floues

Il est maintenant bien connu que, dans de nombreuses situations du monde réel, l'additivité n'est pas une propriété appropriée pour les fonctions d'ensemble, cela à cause de l'absence d'additivité dans de nombreuses facettes du raisonnement humain. Pour pouvoir exprimer la subjectivité humaine (et améliorer la modélisation et l'exploitation des préférences de l'utilisateur), Sugeno proposa de remplacer la propriété d'additivité des fonctions d'ensembles par la monotonie et appela ces mesures monotones non additives des mesures floues.

Considérons l'ensemble des n indices $N = \{1, \dots, n\}$. Selon les applications considérées, ces indices peuvent représenter des attributs, des critères, des juges, des experts, des votants, etc.

Définition : Une mesure floue sur N est une fonction d'ensemble $\mu : 2^N \rightarrow [0, 1]$ qui est monotone, c'est-à-dire $\mu(S) \leq \mu(T)$ chaque fois que $S \subseteq T$, et vérifie les conditions limites $\mu(\emptyset) = 0$ et $\mu(N) = 1$.

« Dans les problèmes d'analyse multicritère, le coefficient $\mu(S)$, pour un $S \subseteq N$ donné, est généralement interprété comme le poids ou l'importance de la combinaison S de critères. Ainsi, en plus des poids usuels sur les critères pris séparément, des poids sur toute combinaison de critères sont également définis. La monotonie signifie alors simplement que le fait d'ajouter un nouveau critère dans une combinaison ne peut faire décroître son importance. » [BOUY et al. 06]

On utilise ainsi le coefficient $\mu(S)$ afin d'obtenir une valeur représentant l'intérêt de l'utilisateur pour un objet étant une combinaison S de critères. C'est cette valeur que l'on va utiliser par la suite pour effectuer des comparaisons entre des objets, afin de déterminer quel est celui le plus intéressant pour l'utilisateur, et à quel point il est intéressant. Grâce à cette valeur, on peut définir un ordre total entre les objets que l'on propose à l'utilisateur.

« À partir d'une telle mesure floue, on peut construire une fonction d'agrégation permettant de calculer une sorte de valeur moyenne en prenant en considération les coefficients de la mesure floue. Une telle fonction d'agrégation est une intégrale floue, concept introduit par Sugeno [SUG 74, SUG 77]. Comme l'intégrale d'une fonction représente généralement sa valeur moyenne, une intégrale floue peut être considérée comme un cas particulier de fonction d'agrégation. » [BOUY et al. 06]

Contrairement aux moyennes arithmétiques pondérées (comme la somme pondérée ou OWA), les intégrales floues sont capables de prendre en considération les interactions éventuelles entre les attributs ou critères. Ces intégrales ont été largement étudiées dans des problèmes d'aide multicritère à la décision [GRA 95a, GRA 96, GRA 98, GRA et al. 00].

« Il existe plusieurs classes d'intégrales floues, parmi lesquelles les plus représentatives sont celles de Choquet et Sugeno. Dans cette section, nous étudierons l'intégrale de Choquet en tant que fonction d'agrégation. En effet, la différence principale entre ces deux intégrales est que la première est appropriée pour agréger des valeurs définies sur une échelle d'intervalle, alors que la seconde est plutôt conçue pour agréger des valeurs définies sur une échelle ordinale. » [BOUY et al. 06]

Ainsi dans le but d'obtenir une représentation flexible des phénomènes complexes d'interactions parmi les attributs ou critères (par exemple, une synergie positive ou négative entre certains critères), il s'est avéré utile de remplacer le vecteur poids par une fonction d'ensemble non additive, permettant ainsi de définir un poids non seulement sur chaque critère, mais aussi sur chaque sous-ensemble de critères.

Par exemple, lors de l'achat d'une voiture, le rapport entre le moteur et sa puissance met en avant des interactions entre chacun des éléments. Il est nécessaire de prendre en compte ces interactions afin de modéliser correctement les préférences utilisateurs.

Une des principales méthodes utilisées pour remplacer le vecteur poids par une fonction d'ensemble non additive correspond aux intégrales de Choquet.

III.3.C. Intégrales de Choquet

La méthode la plus utilisée dans les systèmes multicritère d'aide à la décision (MCDA) est la méthode de somme pondérée, et ce pour des raisons évidentes de simplicité (défini à III.2.B.i).

Définition (rappel)

$$\psi(a_1, \dots, a_n) = \sum_{i=1}^n w_i a_i \quad (18)$$

où les $w_i \in [0, 1]$ sont des poids, et a_i les scores des critères, tels que $\sum_{i=1}^n w_i = 1$

Cette méthode est simple et connue de tous, mais elle présente de nombreuses limites.

Une illustration de ces limites est la suivante (ceci est une adaptation de ce qui a été montré dans [GRA 06]) :

Considérons un problème MCDA à deux critères (critère 1 et critère 2), et 3 objets a , b , c , dont les scores sur les critères sont :

$$u_1(a) = 0.4 \quad u_1(b) = 0 \quad u_1(c) = 1$$

$$u_2(a) = 0.4 \quad u_2(b) = 1 \quad u_2(c) = 0$$

Avec $u_1(a)$ étant le score du critère 1 de l'objet a .

En supposant que les scores sont donnés sur une échelle de 0 à 1.

Le décideur a comme préférence :

$$a > b \geq c$$

Cherchons w_1, w_2 (respectivement, le poids du critère 1 et celui de l'objet 2) tel que ce choix soit représenté par la somme pondérée.

En appliquant la somme pondérée aux 3 objets, on obtient :

$$a \Leftrightarrow 0.4 * w_1 + 0.4 * w_2 \Leftrightarrow 0.4 * (w_1 + w_2)$$

$$b \Leftrightarrow 0 * w_1 + 1 * w_2 \Leftrightarrow w_2$$

$$c \Leftrightarrow 1 * w_1 + 0 * w_2 \Leftrightarrow w_1$$

Si, conformément aux préférences du décideur, on se place dans le cadre où $b \sim c$ ($u(b) = u(c)$), ce qui correspond à dire que le score global de b est équivalent au score global de c , et qu'ainsi donc, on peut en déduire que $w_1 = w_2$ (avec $b \Leftrightarrow w_2$ et $c \Leftrightarrow w_1$).

Si l'on poursuit le raisonnement, on obtient :

$$a > b \Leftrightarrow 0.4(w_1 + w_2) > w_2 \Leftrightarrow 0.4(2 * w_2) > w_2$$

Et l'on obtient donc : $0.8 w_2 > w_2$, ce qui est impossible.

Comment expliquer cette contradiction ? Pour cela, il faut comprendre la signification des poids w_1, w_2 .

Il est bien connu qu'un poids sur un critère n'a pas de sens en soi, mais seulement dans un modèle donné. Pour la somme pondérée, w_1 est en fait le score global d'un objet ayant un score totalement satisfaisant (1) sur le premier critère, et inacceptable (0) sur les autres.

Cependant, notre décideur est plus satisfait par un objet jugé de façon égale sur les deux critères, même si ce jugement reste moyen, que par un objet présentant une évidente faiblesse sur un des deux critères. Il serait possible de tenir compte de cette préférence bien naturelle en considérant non pas que des poids sur les critères pris individuellement, mais aussi des poids définis pour des groupes de critères.

Dans notre cas très simple à deux critères, cela revient à introduire un poids w_{12} sur les deux critères considérés ensemble, et nous gardons comme interprétation que w_{12} est le score global attribué à un objet étant totalement satisfaisant sur les deux critères. Cet objet étant par conséquent le plus satisfaisant possible (car seulement deux critères), il convient de lui donner le score maximal, soit 1, on obtient ainsi :

$$u_1(\text{objet}) = 1$$

$$u_2(\text{objet}) = 1$$

$$u(\text{objet}) = 1 * w_{12} = 1$$

Avec $w_{12} = 1$, et *objet*, un objet différent de a, b ou c .

Afin de modéliser le fait que le décideur considère un objet ne satisfaisant qu'un des deux critères, comme peu acceptable, nous pourrions donner à w_1, w_2 une même valeur assez faible, 0.3 par exemple ($w_1 = 0.3$ et $w_2 = 0.3$).

Essayons maintenant de redéfinir la somme pondérée en tenant compte du nouveau poids w_{12} . En s'en tenant à l'interprétation des poids ci-dessus, il est facile de calculer les scores globaux de a, b, c :

Avec :

$$u_1(a) = 0.4 \quad u_1(b) = 0 \quad u_1(c) = 1$$

$$u_2(a) = 0.4 \quad u_2(b) = 1 \quad u_2(c) = 0$$

« a » a ses scores égaux sur les deux critères, cela correspond donc à la situation représentée par w_{12} , au facteur 0.4 près. En supposant la propriété d'homogénéité du modèle, on pose donc comme score global :

$$u(a) = 0.4 * w_{12} = 0.4 * 1 = 0.4$$

Avec $w_{12} = 1$

« b, c » correspondent respectivement aux situations décrites par w_1, w_2 .

On pose donc :

$$u(b) = w_1 * 0 + w_2 * 1 = 0 * 0.3 + 1 * 0.3 = 0.3$$

$$u(c) = w_1 * 1 + w_2 * 0 = 1 * 0.3 + 0 * 0.3 = 0.3$$

Avec $w_1 = 0.3$ et $w_2 = 0.3$

Les préférences du décideur sont modélisées. En effet, $u(a) > u(b) = u(c)$, et donc $a > b \geq c$.

On voit que pour w_1, w_2 il aurait suffi de prendre n'importe quel chiffre entre 0 et 0.4 exclus. On voit aussi qu'il serait facile de modéliser n'importe quelle préférence entre a, b, c avec cette méthode. On peut cependant arguer que ce cas était extrêmement simple, car les scores correspondaient exactement aux situations décrites par les poids.

Prenons alors un exemple plus compliqué :

Considérons un objet d dont les scores sont :

$$u_1(d) = 0.2$$

$$u_2(d) = 0.8$$

On peut supposer que notre décideur préférera d à b et c , mais sans doute préférera-t-il toujours a à d :

$$a > d > b \geq c$$

En fait, on peut considérer que notre objet d est la somme de deux objets fictifs d' , d'' définis par les scores suivants :

$$u_1(d') = 0.2 \quad u_1(d'') = 0$$

$$u_2(d') = 0.2 \quad u_2(d'') = 0.6$$

On a donc bien :

$$u_1(d) = u_1(d') + u_1(d'') = 0.2 + 0 = 0.2$$

$$u_2(d) = u_2(d') + u_2(d'') = 0.2 + 0.6 = 0.8$$

En supposant notre modèle linéaire, le score global de d sera la somme de scores de d' et d'' . Or nous pouvons calculer ces derniers, car ils correspondent à des situations décrites par des poids.

« d' » a ses scores égaux sur les deux critères, cela correspond donc à la situation représentée par w_{12} , au facteur 0.2 près. En supposant la propriété d'homogénéité du modèle, on pose donc comme score global :

$$u(d') = 0.2 * w_{12} = 0.2 * 1 = 0.2$$

Avec $w_{12} = 1$

Ainsi nous obtenons :

$$u(d') = 0.2 * w_{12} = 0.2$$

$$u(d'') = 0 * w_1 + 0.6 * w_2 = 0.18$$

$$u(d) = u(d') + u(d'') = 0.2 + 0.18 = 0.38$$

Avec, comme dit précédemment : $w_{12} = 1$ et $w_1 = 0.3$ et $w_2 = 0.3$.

Remarquons que nous obtenons l'ordre de préférence désiré :

$$a > d > b \geq c \quad (0.4 > 0.38 > 0.3 \geq 0.3).$$

Cette méthode pour calculer le score global n'est en fait rien d'autre que l'intégrale de Choquet, et les poids sur les groupes de critères définissent une capacité ou mesure floue.

Dans la suite, $N := \{1, \dots, n\}$ désigne un ensemble de critères. On commencera par définir la notion de capacité comme définie par Choquet en 1953 où la notion de capacité généralise la notion de vecteur de poids.

Une fonction $\mu : P(N) \rightarrow \mathbb{R}$ est une *capacité* sur N si $\mu(\emptyset) = 0, \mu(N) = 1$ pour tout $S, T \subseteq N, S \subseteq T \Rightarrow \mu(S) \leq \mu(T)$

Dans le contexte de l'agrégation, $\mu(S)$ peut être vu comme le poids ou l'importance du sous-ensemble $S \subseteq N$ de critères dans la décision.

« L'intégrale de Choquet (discrète) de $x = (x_1, \dots, x_n) \in \mathbb{R}^n$ par rapport à une capacité μ sur N est définie par :

$$\mathcal{C}_\mu(x) := \sum_{i=1}^n x_{(i)} [\mu(A_{(i)}) - \mu(A_{(i+1)})] \quad (19)$$

où on a une permutation sur N telle que

$$x_{(1)} \leq \dots \leq x_{(n)}$$

$$A_{(i)} := \{(i), \dots, (n)\}$$

et ce pour tout $i \in N$, et $A_{(n+1)} := \emptyset$.

Par exemple, si on a $x_3 \leq x_1 \leq x_2$, nous avons :

$$\begin{aligned} \mathcal{C}_\mu(x_1, x_2, x_3) &= x_3[\mu(\{3, 1, 2\}) - \mu(\{1, 2\})] \\ &\quad + x_1[\mu(\{1, 2\}) - \mu(\{2\})] \\ &\quad + x_2\mu(\{2\}) \end{aligned}$$

Où, rappelons-le, $\mu(S)$ peut être vu comme le poids ou l'importance du sous-ensemble $S \subseteq N$ de critères dans la décision. » [GRA 02]

On voit ainsi que la capacité de μ utilise de manière effective les interactions entre les critères. Dans l'exemple précédent on voit en effet que l'on prend en compte le poids du singleton $\mu(\{2\})$, mais également l'interaction de ce singleton avec un autre, comme représenté par $\mu(\{1, 2\}) - \mu(\{2\})$, ou même des interactions de sous-ensembles comme $\mu(\{3, 1, 2\}) - \mu(\{1, 2\})$. On obtient ainsi une capacité prenant en compte toutes les interactions de critères ou d'attributs entre eux, mais aussi de groupe de critères ou attributs.

En tant qu'opérateur d'agrégation, l'intégrale de Choquet satisfait des propriétés particulièrement remarquables ([GRA 97], [MAR 00a]) et a pour cas particulier :

- La moyenne arithmétique pondérée (capacité additive)
- Les combinaisons linéaires de statistiques d'ordre (capacité cardinale)
- Les polynômes laticiels (capacité binaire, qui sont des intégrales de Sugeno définies à partir de mesures floues prenant leurs valeurs dans $\{0, 1\}$)
- En particulier, il a été montré dans [GRA 95b] que la fonction OWA n'est rien d'autre qu'une intégrale de Choquet par rapport à une mesure floue cardinale, c'est-à-dire qui ne dépend que du cardinal des sous-ensembles.

Nous introduisons deux notions qui sont importantes pour une analyse sémantique de l'intégrale de Choquet et des opérateurs couverts par celle-ci. Ce sont l'importance globale d'un critère et l'interaction entre critères.

« L'importance globale d'un critère peut se définir par analogie avec la théorie des jeux coopératifs, où le joueur tient le rôle du critère. Par définition $\mu(\{i\})$ traduit l'importance du critère i . Cependant, il se pourrait que cette valeur soit presque nulle, et pourtant que chaque fois que i se joint à une coalition $A \subset N$, l'importance de cette coalition s'en trouve

sensiblement augmentée, ce qui tendrait à vouloir dire que i est un critère important. Cette idée se trouve parfaitement capturée par la notion d'*indice de Shapley* ou valeur de *Shapley* [SHA 53], issue de la théorie des jeux coopératifs. » [GRA 02]

Cela reflète l'idée de critère inutile lorsqu'il est isolé, mais en groupe il devient utile. Cela fait relativement pensée aux brevets inutilisés jusqu'à ce que de nouveaux soient déposés, et qu'ils deviennent alors indispensables (cf les brevets sur les nouvelles technologies).

Pour tout critère i , son indice de Shapley est défini par [FOD 94] :

$$\phi_i = \sum_{K \subset N \setminus i} \frac{(n - |K| - 1)! |K|!}{n!} [\mu(K \cup \{i\}) - \mu(K)] \quad (20)$$

« La valeur de Shapley est alors le vecteur $[\phi_1 \dots \phi_n]$. L'indice de Shapley ϕ_i calcule la contribution moyenne du critère i dans toutes les coalitions, cette moyenne étant pondérée par un coefficient tenant compte de la cardinalité de la coalition.

La valeur de Shapley est caractérisée de façon unique par un ensemble de quatre axiomes, parmi lesquels $\sum_{i=1}^n \phi_i = \mu(N) = 1$, de telle sorte que la somme des degrés d'importance soit une constante.

On peut remarquer que $\phi_i = \mu(\{i\})$ pour une mesure additive, et que ϕ_i est constant pour une mesure cardinale. » [GRA 02]

Cette valeur permet de déterminer si une information est utile ou non, et permet ainsi de réduire le bruit dans les données. Elle permet également, par exemple, de mesurer l'importance d'éléments, comme des conseillers, dont il est difficile d'évaluer l'impact. Cette mesure permet ainsi d'effectuer cette mesure.

« Le second concept important est l'idée d'*interaction* entre deux critères. On peut en effet imaginer les trois situations suivantes, pour deux critères i, j où leur interaction est notée I_{ij} .

L'importance de i et j pris ensemble, soit $\mu(\{i, j\})$, est plus grande que la somme des importances individuelles $\mu(\{i\}) + \mu(\{j\})$. Dans ce cas, il y a *synergie* ou *complémentarité* entre les critères i et j .

$\mu(\{i, j\})$ est plus petite que la somme des importances individuelles. Cela signifie qu'il y a *redondance* ou *synergie négative* entre i et j .

$\mu(\{i, j\}) = \mu(\{i\}) + \mu(\{j\})$. Dans ce cas, on dit que les critères sont indépendants. » [GRA 02]

Dans le cas d'une synergie, ce concept capture l'idée de l'expression bien connue « $1 + 1 = 3$ ». C'est-à-dire que le résultat de la somme de ce qui est produit par deux éléments est supérieur à ce qu'auraient produit ces deux éléments pris individuellement. Ainsi on peut connaître des tuples d'éléments, qui lorsqu'ils sont réunis, satisfont l'utilisateur. La synergie négative représente ainsi l'inverse et permet de détecter des critères que l'utilisateur ne trouve pas intéressant ou pire qu'il n'aime pas.

Avec la formule suivante pour l'importance globale pour toutes les coalitions possibles, on obtient la définition suivante pour l'indice d'interaction [MUR 93] :

$$I_{ij} = \sum_{K \subset N \setminus \{ij\}} \frac{(n - |K| - 2)! |K|!}{(n - 1)!} [\mu(K \cup \{i, j\}) - \mu(K \cup \{i\}) - \mu(K \cup \{j\}) - \mu(K)] \quad (21)$$

Et en voici la version définie pour un nombre quelconque de critères [GRA 97] :

$$I(A) = \sum_{K \subset N \setminus A} \frac{(n - |K| - |A|)! |K|!}{(n - |A| + 1)!} \sum_{B \subset A} (-1)^{|A| - |B|} \mu(K \cup B), \forall A \subset N \quad (22)$$

« On a alors $\phi_i = I(\{i\})$ et $I_{ij} = I(\{i, j\})$. Ceci permet de définir la notion de *mesure k-additive*, qui est telle que les indices d'interactions sont nuls pour tout sous-ensemble de plus de k éléments. Ainsi, une mesure 1-additive est une mesure additive au sens ordinaire.

L'expression de l'intégrale de Choquet en fonction des indices de Shapley et d'interaction est possible pour les mesures 2-additives (i.e. qui peuvent être définies uniquement avec ces indices). » [GRA 02]

Ce qui correspond, pour l'intégrale de Choquet à considérer uniquement des couples de critères ou attributs. Ce qui est suffisant, étant donné que le fonctionnement humain habituel correspond la plupart du temps à associer des critères ou attributs par couple.

L'expression de l'intégrale de Choquet en fonction des indices de Shapley et d'interaction est la suivante :

$$\mathcal{C}_\mu(a) = \sum_{I_{ij}>0} (a_i \wedge a_j) I_{ij} + \sum_{I_{ij}<0} (a_i \vee a_j) |I_{ij}| + \sum_{i=1}^n a_i (\phi_i - \frac{1}{2} \sum_{j \neq i} |I_{ij}|) \quad (23)$$

« Cette expression montre bien le sens des indices d'interaction dans l'agrégation des critères :

- si l'interaction est positive, l'agrégation est conjonctive (min) ;
- si l'interaction est négative, l'agrégation est disjonctive (max) ;
- l'indice de Shapley est la partie linéaire de l'intégrale de Choquet.

» [GRA 02]

III.4. Conclusion

Nous avons ainsi détaillé de nombreuses méthodes permettant d'apprendre les préférences d'un utilisateur, et cela en prenant ou non en compte les interactions entre les critères. Ces méthodes permettent de faire évoluer le profil de l'utilisateur au fil de ses utilisations du système mettant en œuvre ces méthodes. Ainsi ces méthodes se démarquent des méthodes traditionnelles en Intelligence Artificielle (IA), qui apprennent un profil, ou une fonction, qui ne va plus varier.

Nous avons également détaillé de nombreux opérateurs. Des opérateurs d'agrégation et de désagrégation ont été présentés. Et cela dans le cas où l'on ne prend pas en compte les interactions entre les critères, et également dans le cas où elles sont prises en compte.

Le choix entre ces méthodes et opérateurs se fera en fonction du système dans lequel ces méthodes et opérateurs vont être utilisés.

Conclusion générale de l'état de l'art

Tout au long de ses trois chapitres, de nombreux concepts et méthodes ont été présentés.

Or ces concepts et méthodes ne permettent pas de mettre en œuvre un système pour assister dans leurs décisions les utilisateurs permettant d'utiliser dynamiquement un nombre variable de critères.

Elles ne permettent pas non plus d'utiliser des profils évolutifs qui permettent aux utilisateurs de changer de préférences à n'importe quel moment et d'ainsi avoir un système qui s'ajuste de manière très rapide aux nouvelles préférences. Par exemple, avoir une préférence pour un auteur de livre, et le lendemain ne plus aimer cet auteur. Pour cela, il faut utiliser des opérateurs de désagrégation particuliers (par exemple, la décomposition additive non transitive) et les insérer au sein de méthode permettant ses adaptations.

De plus, il est également nécessaire de mettre en œuvre un système mettant en œuvre les principes précédemment détaillés et avec lesquels on utilise les informations issues des interactions entre les critères. Cela afin d'utiliser un maximum d'information pertinente pour effectuer des recommandations les plus efficaces.

Nous avons ainsi mis en place un système cohérent, afin de répondre à ces problèmes, et avons développé les méthodes associées nécessaires.

Partie II. Contributions

Les contributions de cette thèse portent tout d'abord sur la définition, la construction et l'évolution d'un profil utilisateur en fonction des actions explicites et implicites de l'utilisateur. Ce profiling évolutif est mis en œuvre au sein d'un système de recommandation utilisable sans base d'apprentissage, de manière synchrone et totalement incrémentale, et qui permet aux utilisateurs de changer rapidement de préférences et même d'être incohérents (rationalité limitée). Ce système, qui vient en complément d'un système de Recherche d'Information, a pour objectif d'établir un ordre total sur une liste d'éléments proposés à l'utilisateur, et ce en concordance avec les préférences de l'utilisateur. Des travaux décrivant une partie du système détaillé dans cette thèse ont déjà paru dans [MART et al. 12a] et [MART et al. 12b].

Le travail contribue également à la définition de techniques qui permettent d'apporter en partie des solutions à des verrous technologiques comme la désagrégation de critères et la prise en compte d'un nombre variable de critères dans le processus d'aide à la décision interactif, et ce sans définir au préalable de famille cohérente de critères sur laquelle est basée la décision.

Nous allons commencer par détailler le système de profiling et de recommandation que nous avons mis en place, puis les méthodes de calcul définies pour effectuer la désagrégation de critères dans le cadre où les critères sont indépendants les uns des autres (indépendance préférentielle), et enfin les méthodes de désagrégation de critères dans le cadre où les critères interagissent les uns avec les autres (synergies négative ou positive).

Chapitre IV. Modèle de prise en compte de profil (profiling)

IV.1. Contexte et contraintes

Le système que nous présentons est un système de recommandation, dont le but est d'assister l'utilisateur dans ses choix en fonction de ses préférences, que nous apprenons et faisons évoluer au fur et à mesure que l'utilisateur utilise notre système. Nous ne nous limitons pas à sélectionner LA solution qui satisfera le plus l'utilisateur. En effet, « La pratique de l'aide multicritère à la décision amène souvent à vouloir bâtir des

recommandations de nature plus variée que le seul choix d'une et d'une seule action » [ROY 85], c'est pour cela que nous proposons une liste de solutions triées à l'utilisateur.

Le système que nous mettons en place vient en complément d'un système de recherche d'informations. En effet, comme annoté au chapitre I.2, le filtrage d'information, dont font partie les systèmes de recommandation, est un processus dual de la recherche d'informations. Ainsi le but de notre système est d'ordonner les solutions qui sont proposées à l'utilisateur, cette liste de solutions étant fournie par un système de Recherche d'Information. Cela afin de correspondre le mieux possible (être le plus proche possible) à l'ordre idéal de ces solutions, celui-ci dépendant des préférences de l'utilisateur.

Plusieurs contraintes ont été définies. En effet, le système sera utilisé par des utilisateurs divers, et dont on ne sait rien. En effet, nous ne disposons pas de base d'informations au préalable sur l'utilisateur et donc on ne peut pas utiliser d'algorithmes utilisant une base d'apprentissage. De plus, les utilisateurs ont la possibilité de changer de préférences, et même de ne pas toujours être rationnels (la rationalité limitée).

Une autre contrainte vient du fait que les informations récupérées sur les objets choisis ou notés par les utilisateurs arrivent de manière séquentielle et en temps réel (nécessité d'un système synchrone). Il nous faut donc apprendre le profil de l'utilisateur au fur et à mesure que les données sont collectées. Il est nécessaire d'apprendre en continu, car aucun apprentissage différé n'est réalisé.

Nous voulons que le système puisse être utilisé sans nécessiter de grosses infrastructures, ainsi, les temps de calcul doivent être acceptables, ce qui implique de ne pas utiliser des calculs trop lourds. C'est-à-dire, que les temps de calcul doivent permettre au système une utilisation en temps réel, sans temps d'attente perceptible pour l'utilisateur (par exemple : *temps de réponse* < 1 seconde). Ainsi, des techniques comme celles mises en œuvres dans les systèmes de recommandation collaborative de type « *user-centric* » ou « *memory-based* » ne peuvent pas être utilisées. En effet, cette méthode est maintenue en permanence en mémoire dans le serveur et utilisée directement pour générer des recommandations à l'utilisateur. Cette approche *memory-based* est théoriquement plus précise, cela étant dû à sa capacité d'avoir en permanence et en temps réel à sa disposition, toutes les données pour générer les recommandations. Cependant, elle souffre de problèmes pour des bases de données de millions d'utilisateurs et de millions de contenus. En effet, souvent, les plates-formes sur lesquelles sont utilisés les filtres collaboratifs ont des millions d'utilisateurs, de produits et contenus. Cela demande donc beaucoup de puissance de calcul pour pouvoir proposer des suggestions aux utilisateurs. Ce problème est connu sous le nom de problème de *scalability*.

Cependant, dans une approche « *item-centric* » ou « *model-based* » (recommandation par contenu), les données sont préalablement traitées hors-ligne (matrice *item-item*), ou lors de l'exécution de l'application ou service web, le profil de l'utilisateur sera directement utilisé pour effectuer les recommandations. L'approche *model-based* permet donc d'éviter le problème de *scalability*.

Le système doit avoir la capacité de pouvoir être utilisé avec n'importe quel type d'objets (tels que des livres, des cartes, des maisons, etc.). On ne peut donc pas utiliser les approches déjà mises au point, liées par exemple à la récurrence d'un mot dans un texte. De plus, les fonctions de décision de type « seuil » que l'on trouve dans la plupart des systèmes de filtrage d'informations et qui consistent, d'une façon générale, à mesurer une similitude entre un élément et un profil (si le score obtenu est supérieur à un seuil défini, alors ce document est sélectionné et sinon il est rejeté), ne peuvent être mises en œuvre étant donné que l'on ne filtre pas les solutions.

IV.2. Modèle proposé

Le modèle que nous présentons ici est un système de recommandation basé sur le contenu, celui-ci met en œuvre un système de profils multicritère évolutif de l'utilisateur, cela à travers une exploitation des données obtenues par les actions que l'utilisateur réalise lors de ses interactions avec le système. Il vient en complément de systèmes de Recherche d'Information, comme celui de Google [BRI 98], que nous ne détaillerons pas, cela n'étant pas l'objectif de nos travaux.

Dans le modèle mis en place, les actions de l'utilisateur permettent d'obtenir des données explicites et implicites qui sont utilisées lors de la mise en œuvre de méthodes d'apprentissage et de désagrégation afin d'effectuer des recommandations à l'utilisateur en fonction de son profil. Cette recommandation prend la forme d'un tri (*ranking*) des solutions proposées à l'utilisateur, grâce à l'utilisation d'une méthode de calcul de score (*scoring*), permettant d'obtenir un score global. Ce score est calculé, pour la solution évaluée, à partir des caractéristiques des solutions proposées et en fonction du profil de l'utilisateur.

Nous commencerons par définir le contexte global dans lequel se situe notre système, à savoir un système de recommandation.

IV.2.A. Système de Recommandation

Nous souhaitons que les profils des utilisateurs puissent être les plus précis possible, afin de refléter au mieux leurs préférences. Cela conduit à des profils uniques pour chaque utilisateur. On ne peut donc pas utiliser des solutions employant des profils génériques, dans lesquels on classe l'utilisateur afin de lui recommander les éléments associés à ces profils (méthode des centroïdes [SOB 06]), puisque les profils des utilisateurs vont naturellement être très précis. En effet, plus les profils seront précis et plus ils seront distants les uns des autres, ainsi que distants de la même manière des centroïdes.

De plus, nous voulons également que le système puisse être utilisé directement, sans utiliser une grande collection de données des actions d'autres utilisateurs, afin d'avoir un système rapidement utilisable et efficace. On ne dispose pas de base d'apprentissage afin d'initialiser le système. Or les systèmes de recommandation basés sur les actions des autres utilisateurs présentent le besoin de nombreuses informations disponibles à l'initialisation afin de pouvoir être efficaces (*cold start*) et posent également des problèmes de *scalability*. Qui plus est, les systèmes de recommandation collaborative présentent des problèmes dits de « rareté » (*Sparsity*). En effet, le nombre de produits ou contenus est énorme sur certaines plates-formes, et même les utilisateurs les plus actifs n'auront noté ou valorisé qu'un tout petit sous-ensemble de toute la base de données. Donc, même l'article le plus populaire n'aura que très peu de bonnes notes. Dans une telle situation, deux utilisateurs auront peu d'articles valorisés en commun, ce qui rend plus difficile la tâche de corrélation. C'est une situation qu'on retrouve lorsque le système dispose d'un ratio élevé de contenu par rapport aux utilisateurs, et qu'on retrouve aussi souvent au stade initial du lancement du service de recommandation, ce qui nous ramène au problème de cold start (qui peut être vu comme un cas spécial du problème de rareté). Nous ne pouvons donc pas utiliser de système se basant sur des techniques de recommandation collaborative, et ainsi, les méthodes hybrides utilisant la recommandation collaborative ne sont pas non plus utilisables.

Pour ce qui est des systèmes comme les systèmes recommandation démographiques, ils ne sont pas adaptés à notre problématique, car nous n'utilisons pas de données liées à des catégories démographiques. Nous ne pouvons pas non plus utiliser de système de recommandation basé sur une valeur d'utilité, car nous ne calculons pas et ne stockons pas de valeur d'utilité globale pour l'utilisateur, mais utilisons le profil évolutif de l'utilisateur. Et enfin, nous ne pouvons pas utiliser de système de recommandation basé sur l'exploitation de connaissance, car les utilisateurs ont la possibilité de changer de préférences, et même de pouvoir ne pas toujours être rationnels. Ainsi, l'utilisation d'inférences logiques stables sans

possibilité d'ajustement ne peut donc être effectuée, on peut cependant remarquer que cela pourrait être possible à un meta-niveau.

Le modèle que nous mettons en œuvre utilise donc une approche sur le contenu (Recommandation Objet). En effet, cette approche ne souffre pas des problèmes décrits précédemment, cependant cette approche est caractérisée par un défaut important, qui est : la quantité, la justesse et la précision des données disponible décrivant les éléments. Afin de résoudre ce problème, la quantité d'informations disponible, décrivant les caractéristiques des objets, a été augmentée en utilisant plusieurs sources de données.

De plus, on remarquera que nous nous différencions des systèmes de recommandations mis en place, par exemple, comme Google Books (*books.google.fr*) ou Amazon (*amazon.fr*, qui favorise la recherche et le référencement de titres, la vente d'exemplaire), d'une part par le choix de n'utiliser uniquement que l'approche par le contenu (pas de système hybride), mais également, car nous proposons des recommandations plus personnelles, qui vont plus en profondeur dans les caractéristiques des objets proposés afin de mieux satisfaire l'utilisateur. Si on considère Google Books, pour l'instant, le moteur fonctionne pour un utilisateur moyen. Or tous les utilisateurs n'ont pas le même profil de recherche : certaines requêtes vont être plutôt basées sur les contenus (lors de la recherche d'un thème ou d'un sujet d'information), d'autres cherchent précisément des titres ou des auteurs. Et ces deux profils de recherche n'attendent pas les mêmes résultats d'une requête. Parfois, les utilisateurs sont à la recherche d'un aperçu. Parfois, ils sont à la recherche d'informations sur cet ouvrage. Parfois encore, ils veulent acheter un exemplaire de ce livre. Le nouvel algorithme mis en place va plutôt aider ceux qui cherchent spécifiquement un titre plutôt que les autres. Finalement, comme le moteur de recommandation d'Amazon, bridé pour satisfaire des nécessités marketing, Google Books est optimisé dans un certain but, avec une certaine vision de ce que nous devons y chercher et y trouver. Les ingénieurs qui manipulent l'algorithme de Google Books ont visiblement décidé de favoriser la recherche et le référencement de titres, la vente d'exemplaire... Notre proposition se démarque ici, car notre but est précisément d'effectuer la recherche dans l'ensemble du corpus et la mise en perspective de mots clefs.

Afin de mieux prendre en compte les préférences des utilisateurs, et pour mieux les assister, nous utilisons un système utilisant des concepts issus des domaines du Multicritère et du Multiattribut. Cela nous permet d'avoir une meilleure gestion et prise en compte des préférences de l'utilisateur. Nous avons donc défini le type des profils comme étant multicritère.

IV.2.B.Profils utilisateurs

Les utilisateurs possèdent plusieurs profils, en fonction de leur localisation (profil à la maison, au travail par exemple), et également par thématique (Livres, Films, etc.). Suivant le contexte d'utilisation de notre système, l'utilisateur peut se voir demander quel profil il faut utiliser. En effet, s'il se trouve sur une plateforme n'utilisant qu'un seul type de profil, on ne demandera pas à l'utilisateur quel profil utiliser, mais dans le cas contraire, nous lui demandons explicitement quel profil utilisé. Le fait de déterminer le profil à utiliser de manière automatique représente une perspective que nous n'avons pas encore explorée. Ainsi, pour la suite, on ne travaille qu'avec un seul profil à la fois, c'est pour cela que l'on ne parle plus que de profil de l'utilisateur et non des profils de l'utilisateur.

Le profil de l'utilisateur est de type multicritère, afin de pouvoir correspondre au maximum à ses préférences, on utilise uniquement les données liées au domaine d'intérêt, et ce afin de pouvoir définir précisément les préférences de l'utilisateur pour l'aider dans ces décisions. En effet, nous ne prenons pas en compte les données personnelles pour des raisons de respect de sa vie privée. L'utilisateur ne contrôlant pas la source de l'information lorsqu'il utilise un système de recommandation, des domaines tels que la préférence de livraison, la sécurité ou encore la qualité, ces données ne sont pas employées dans les profils des utilisateurs.

Le principe de mettre en place une ontologie personnelle est très tentant, car ainsi on dispose d'un profil très précis et très personnel vis-à-vis de l'utilisateur. Cependant, cela oblige l'utilisateur à un très grand investissement afin de la construire. Or nous ne souhaitons pas cela, car nous voulons éviter au maximum à l'utilisateur d'avoir à donner explicitement des informations, afin que le système soit plus facile à utiliser. C'est pourquoi nous utilisons des vecteurs d'attributs pondérés décrits de la manière suivante :

Définition d'un vecteur d'attributs pondérés :

Soit V un vecteur d'attributs pondérés décrivant un profil utilisateur, ce vecteur est décrit par un ensemble d'attributs $\{a_1, \dots, a_n\}$ composant V et également un score s_i de l'attribut a_i avec $n \geq 1, n \in \mathbb{N}$ et $i \in [1, \dots, n]$

$$V = \{as_1, \dots, as_n\} \text{ avec } as_i = (a_i, s_i) \text{ avec } as_i \text{ un terme } a_i \text{ de } V \text{ et un score } s_i$$

Exemple :

$$V = \{(photographie, 8), (art, 5), (couleur, 3), (caméra, 7)\}$$

$$V = \{(voiture, 9), (moteur, 2), (puissance, 7), (course, 8), (couleur, 6)\}$$

On remarquera que l'on utilise ici des scores et non des poids, et que les scores pour un même attribut peuvent varier entre 2 profils. Ainsi, pour chaque profil, la somme des scores n'est pas égale à 0, on ne normalise pas. L'utilisation de score permet en effet une plus grande amplitude dans les écarts des préférences des utilisateurs et permet une plus grande réactivité aux changements de préférences des utilisateurs.

Cette formulation est équivalente à celle donnée au chapitre I.1.B de vecteur de terme pondéré, mais permet d'avoir une meilleure visualisation du vecteur et d'utiliser des scores à la place d'une pondération.

Les assistants personnels sont une extension naturelle des systèmes de recommandations, ces systèmes utilisant un ou plusieurs profils. Notre système se rapproche de ces assistants, à la différence près que, contrairement à la plupart des assistants personnels, nous ne possédons pas de base d'apprentissage.

Les profils sont donc constitués de termes valués en fonction des actions et réponses de l'utilisateur sur les éléments qui lui sont présentés. Des méthodes (décrites plus loin), viennent mettre à jour de manière automatique ces valeurs contenues dans les profils de l'utilisateur.

On ne propose pas à l'utilisateur de gérer lui-même ses préférences. « Afin de pouvoir exprimer ses préférences, l'utilisateur doit être capable de comprendre la signification de chaque facteur et des métriques utilisées pour l'évaluer. » [BOUZ et al. 07]. En effet, il arrive fréquemment que les utilisateurs définissent des préférences et que par la suite, leurs actions démontrent le contraire. Ainsi, c'est uniquement le système que nous proposons qui se charge de la répartition de l'importance de chaque critère du profil de l'utilisateur. Nous acceptons l'idée qu'un utilisateur puisse changer d'avis ou bien soit incohérent.

On remarque qu'il serait également intéressant d'utiliser les données issues d'un profil dans un autre. Par exemple, on pourrait utiliser des données d'un profil sur des livres au sein d'un profil sur des films. En effet, ces deux profils partagent des données comme les genres/styles que l'utilisateur aime. Cependant, étant donné que les critères/attributs utilisés dans chaque profil varient et ne respectent pas une forme précise, il faudrait mettre en place de manière manuelle des transferts d'informations entre les profils. Cela représente une autre perspective que nous n'avons pas encore explorée.

Dans le système que nous mettons en place, la valeur des critères au sein des profils utilisateurs ne fait que croître, et ce sans borne supérieure. C'est un choix volontaire, car nous ne partons pas du principe où il faut diminuer la valeur d'un critère lorsqu'il est peu présent dans les éléments sélectionnés par l'utilisateur. En effet, nous ne prenons en compte

que les données fournies explicitement ou indirectement par l'utilisateur, et ne faisons pas d'inférences à partir de ses données. Ainsi si l'utilisateur n'a pas donné d'informations à propos d'un critère, alors nous ne faisons pas évoluer la valeur de ce critère dans le profil de l'utilisateur. De même, comme dit précédemment, les utilisateurs sont habitués à des échelles de notes positives, c'est pourquoi, afin d'être cohérents, nous utilisons des échelles de notes positives.

Nous pouvons ainsi construire des profils adaptés et nous en servir au sein de notre système de recommandation afin d'effectuer des suggestions à l'utilisateur. Cette recommandation intervient dans le cadre d'un filtrage d'informations.

IV.2.C. Filtrage d'information

Lors d'une opération de filtrage d'informations, de nombreux modèles de filtrage ont recours à une collection d'entraînement qui impose à l'utilisateur une phase initiale de travail. Ceci pose une première limite, car dans une situation courante de recherche/filtrage incrémental d'information, nous ne disposons d'aucune connaissance sur les informations à rechercher/filtrer.

« De plus, la plupart des méthodes d'apprentissage de profils se basant sur des échantillons de documents sont dépendantes de la taille de l'échantillon. Il a été montré notamment dans les travaux de Zhang [ZHAN 04] que les méthodes basées sur la régression logistique sont plutôt performantes lorsque le nombre de documents de l'échantillon est conséquent. » [TEB 04]

Pour pallier à ces deux inconvénients, et résoudre les contraintes définies précédemment, nous proposons une méthode d'apprentissage purement incrémentale, avec une approche basée sur le contenu, qui ne nécessite aucune connaissance sur l'utilisateur, au démarrage du processus de filtrage. Cet apprentissage utilise des données (critères valués), issues d'une méthode que nous proposons afin de réaliser la désagrégation des informations des objets présentés à l'utilisateur et sur lesquels l'utilisateur a effectué une action (mettre une note, ou sélectionner un élément). Le système d'aide à la décision mis en œuvre est interactif, ainsi les données du profil sont mises à jour au fur et à mesure que le système reçoit des informations (implicites ou explicites), extraites des actions de l'utilisateur. Ce fonctionnement permet également d'apprendre les profils d'une manière uniforme tout au

long du processus, comparativement aux méthodes basées sur Rocchio [ROC 71] ou sur la régression logistique.

Pour le choix du moment auquel mettre le système à jour, la solution retenue est celle de la solution synchrone. En effet, nous voulons que le système se mette à jour à chaque objet reçu (informations déduites d'une action implicite ou explicite de l'utilisateur). C'est-à-dire de manière adaptative et incrémentale. Les valeurs des critères sont déduites à partir des objets rencontrés cumulés dans le temps.

On remarquera que nous n'utilisons pas de méthode permettant de construire une matrice de corrélation des objets entre eux, pour, par exemple, recommander un objet proche de ceux que l'utilisateur a aimés. En effet, nous avons mis en œuvre une méthode de scoring qui rend inutile la construction d'une telle matrice. Cette méthode de scoring permet d'obtenir directement un score de pertinence pour l'utilisateur en examinant les caractéristiques des éléments, et nous permet ainsi d'éviter un coût de calcul important en définissant la matrice de corrélation et en la maintenant à jour.

Les objets rencontrés par l'utilisateur, sur lesquels l'utilisateur doit réaliser une action, sont utilisés dans le cadre de diverses méthodes afin d'en extraire des données, qui sont utilisées afin d'apprendre les préférences de l'utilisateur.

IV.2.D. Apprentissage des préférences

Parmi les principales méthodes pour réaliser de l'apprentissage de préférences, nous disposons des réseaux de neurones [FRA 96] et de l'apprentissage symbolique automatique (par exemple via un système d'apprentissage par renforcement).

Or l'inconvénient majeur des réseaux de neurones est lié à leur aspect « boîte noire », car la fonction mathématique représentée par le réseau de neurones devient vite trop complexe pour l'analyser et la comprendre directement.

Des raisons sont exposées par [SLO 94] pour utiliser une approche par apprentissage symbolique automatique plutôt qu'une approche neuronale. En effet l'apprentissage symbolique automatique est souvent mieux adapté aux problèmes de classification qu'un apprentissage numérique. C'est le cas par exemple dans [WEI 91], où des méthodes d'apprentissage symbolique automatique sont comparées avec le type de réseaux de neurones actuellement le plus utilisé, celui de rétro-propagation [RUM 86]. Les performances du réseau neuronal étaient pratiquement toujours moins bonnes. Il faut néanmoins constater

que le réseau de rétro-propagation n'est pas toujours le réseau neuronal le plus adapté aux problèmes de classification simples. Ses capacités de régression non linéaire deviennent souvent utiles uniquement pour des problèmes assez complexes. Les résultats obtenus dépendent également beaucoup de la nature même des variables de paramétrage utilisées au sein du réseau.

Il existe d'autres méthodes, notamment des algorithmes génétiques (en anglais : genetic algorithms) [HOLL 75], [DE JONG 75] et des méthodes de raisonnement à base de cas (en anglais : case-based learning) [KOL 87], ainsi que des réseaux Bayésiens [DAN 07]. Néanmoins l'utilité de ces méthodes pour notre problématique ne semble pas évidente. Les algorithmes génétiques permettent avant tout de faire de l'optimisation [GRE, 85, 87]. Le raisonnement à base de cas souffre des mêmes difficultés à traiter des valeurs quantitatives que les bases de règles et peut poser des problèmes de construction. Les réseaux Bayésiens sont utilisables, mais les préférences d'un utilisateur sont souvent floues et leur traduction en un modèle probabiliste numérique est très difficile.

La méthode qui nous intéresse le plus est celle dite par renforcement. En effet cette méthode se prête particulièrement bien à la mise à jour de profils. En effet, « L'apprentissage par renforcement est une approche informatique de l'apprentissage dans laquelle un agent essaie de maximiser le montant total de la récompense qu'il reçoit en interagissant avec un environnement complexe et incertain » [SUT 98]. Dans le système que nous mettons en œuvre, l'utilisateur (l'agent) va maximiser le total de la récompense (la qualité des recommandations qui lui sont faites), par ses interactions. Grâce à ces actions itératives, le profil de l'utilisateur va être mis à jour de manière synchrone au fil des actions de l'utilisateur, et des données extraites et exploitées de ces mêmes actions. De plus, l'apprentissage automatique est souvent combiné avec des méthodes d'acquisition de connaissances (les actions itératives dans notre cas), qui facilitent l'interaction avec le décideur (voir par exemple dans [SEN 95]).

C'est pour cela que nous la mettons en œuvre dans notre système. De plus la mise en œuvre d'un apprentissage par renforcement appliquée aux profils utilisateurs a déjà été expérimentée avec succès dans le cadre des travaux de recherche de Tmar [BOUG et al. 02], [TMA 02]. C'est sur ces travaux que nous nous basons. Cependant, une adaptation de ces travaux a été faite afin d'adapter cette méthode à notre but et à notre type de données qui ne sont pas des textes, mais des éléments (des objets par exemple). Dans notre système, les données utilisées sont diverses, et varient en fonction des éléments et de leur description. De plus, à partir des données mises en œuvre, il n'existe pas de méthodes permettant de se ramener à des données de type texte uniquement.

Dans cette adaptation, nous utilisons l'approche « agréger puis comparer » (AC) décrite dans [GRA 02] qui est particulièrement adaptée à notre cas pour effectuer la comparaison entre les différentes alternatives. En effet, cette approche est sensée résumer la valeur globale d'une alternative et sert de base à la comparaison multicritère des alternatives.

Mais nous utilisons également une variation de l'approche « comparer puis agréger » (CA) [GRA 02] pour comparer des alternatives entre elles, et ensuite agréger des données extraites de cette comparaison avec d'autres données. Cette variation a été mise en place dans l'algorithme dit « Pairwise » décrit plus loin.

Afin de mieux visualiser le fonctionnement global du système que nous avons mis en place, nous définissons l'algorithme général utilisé.

IV.2.E. Algorithme général

La méthode utilisée consiste tout d'abord à construire un vecteur temporaire de critères valués à partir des critères composant un objet noté ou choisi par l'utilisateur (ce sont les deux actions possibles qui sont à dispositions de l'utilisateur pour faire évoluer son profil). Ce profil temporaire est ensuite intégré dans le profil global de l'utilisateur. Ce profil global est ensuite utilisé, afin de calculer un score, correspondant aux préférences de l'utilisateur, pour chaque objet qui est évalué. Ces scores vont ainsi être utilisés pour présenter ces objets à l'utilisateur de manière triée en fonction des scores calculés et de manière descendante.

On remarque que dans le modèle présenté ici, un classement peut toujours être obtenu, en effet on ne compare pas les critères entre eux. Il n'y a pas de problème de Pareto optimalité, car l'on utilise une méthode d'agrégation liée à la théorie Multiattribut, nous permettant de calculer un score pour chaque objet sans avoir à les comparer entre eux.

Le processus général se décompose ainsi :

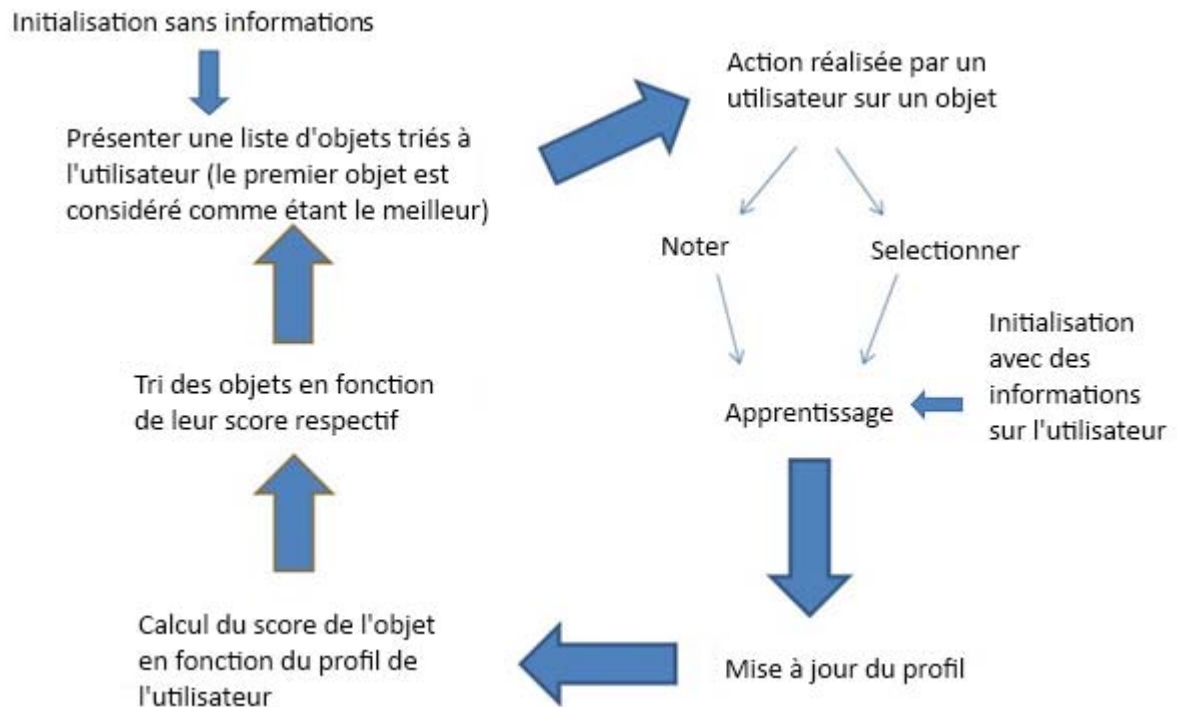


Figure 8. *Fonctionnement général du système*

Comme on le voit sur la Figure 7, lors de l'initialisation, on commence par présenter une liste aléatoire d'objets à l'utilisateur s'il n'a pas donné d'informations et l'on commence à l'étape où l'utilisateur effectue une action sur un objet. Dans le cas où l'utilisateur a fourni des informations sur ses préférences, on suit le processus et l'on arrive à l'étape où l'utilisateur effectue une action sur un objet.

- 1) L'utilisateur commence par faire une action sur un objet d'une liste d'objets qu'on lui soumet. En effet, s'il connaît déjà un objet dans la liste, il va donner une note à cet objet, ce qui revient pour nous à donner ses préférences. Sinon, s'il ne connaît aucun objet dans la liste, il regarde les caractéristiques des objets et choisit celui qu'il préfère. L'action de choisir est moins forte que l'action de noter. En effet, l'action de choisir est basée uniquement sur la description de l'objet alors que l'action de noter est basée sur la connaissance et l'expérience de l'utilisateur. Ainsi l'impact de noter, dans la mise à jour du profil, sera plus grand que celui de choisir. Dans le cas de la note, cela correspond à des données explicites, et dans le cas du choix, il s'agit de données implicites. Or les données explicites, de par leur nature, car non

interprétées, sont plus sûres que celles implicites, et donc leur impact est plus important.

- 2) À partir de son choix, un calcul en fonction de la note (algorithme n°1) ou un calcul de type Pairwise (algorithme n°2) est lancé dans le but de désagréger les performances de l'objet.
- 3) Avec ces données obtenues, le système va mettre à jour le profil de l'utilisateur.
- 4) Le système va effectuer un calcul de score pour chaque objet disponible (toute la base de données, ou une sous-partie), basé sur le profil de l'utilisateur.
- 5) On trie les objets en fonction de ces nouveaux scores.
- 6) On présente cette nouvelle liste à l'utilisateur et on recommence à l'étape 1. On remarquera que cette liste est limitée à 10 éléments, en effet la majorité des utilisateurs ne vont pas regarder les éléments au-delà du 10ème [INS 09].

En effet, tout l'enjeu ici est de définir le poids de chaque attribut dans le profil de l'utilisateur, de façon à ce que lorsque l'on calcule le score des objets, en fonction de son profil, on obtienne un ordre d'objets qui corresponde aux préférences de l'utilisateur.

Le système est interactif, en effet on alterne les phases où l'utilisateur fournit des informations (de manière implicite ou explicite), et les phases où l'on utilise ces données afin de faire de nouvelles recommandations à l'utilisateur.

Voici un exemple d'utilisation de la plateforme en ligne (*mobifiz.fr*) encore au stade de bêta où l'on a mis en place le système décrit dans cette thèse. Cela permet de mieux appréhender le travail effectué dans cette thèse et l'interactivité avec l'utilisateur. En effet, on peut y voir des notes fournies par l'utilisateur (Appréciation) et les recommandations faites après que l'utilisateur ait joué à un quiz (des données implicites ont été fournies lors de son choix de quiz). L'interactivité s'expose entre les phases où le système propose des recommandations sous forme de liste de quiz trié, et entre les phases où l'utilisateur sélectionne, note et joue à un quiz.

On propose à l'utilisateur une liste de quiz qui correspondent à ses préférences, ici dans le domaine du sport. Les quiz en rouge correspondent aux quiz auxquels l'utilisateur a déjà joué. On constate qu'à ce moment-là, l'utilisateur aime les quiz sur le cyclisme, et l'utilisateur a déjà joué à des quiz sur l'athlétisme et le cyclisme.



Figure 9. Exemple d'interface en utilisation réelle avec Mobifiz

Un utilisateur sélectionne un quiz en athlétisme dans la liste de quiz (Figure 9). L'utilisateur a effectué les actions de sélectionner et de jouer à un quiz sur l'athlétisme. Notre système a pris en compte ces informations et a mis à jour le profil de l'utilisateur. Le système a par la suite effectué un nouveau tri des quiz disponible pour l'utilisateur. Voici la nouvelle liste affichée :



Figure 10. Exemple d'interface en utilisation réelle avec Mobifiz après interaction de l'utilisateur

Sur la figure 10, on constate aisément qu'il y a de nombreux quiz qui concernent l'athlétisme dans les premiers résultats affichés. Ainsi on peut constater l'efficacité et l'utilité de notre système utilisé de manière directe sur la plateforme en ligne.

IV.3. Conclusion

Le système proposé met ainsi en œuvre diverses techniques, où leur utilisation est justifiée par rapport aux contraintes qui ont été définies à savoir, un système évolutif, multicritère/multiattribut, totalement incrémental (synchrone), qui n'utilise pas de base de données a priori, et où les utilisateurs peuvent changer de préférences.

Le système proposé permet ainsi d'apporter une solution à chaque contrainte qui a été formulée et permet de fournir aux utilisateurs une assistance à leurs prises de décision, et ce en fonction de leurs préférences.

Une perspective à notre travail serait d'utiliser les scores présents dans les profils des utilisateurs afin de déterminer quels sont les vrais critères, quasi-critère et pseudo-critère. Cela permettant ainsi d'offrir une visualisation des préférences de l'utilisateur.

Cependant, les méthodes de désagrégation actuellement présentes ne conviennent pas. En effet, seule la méthode de décomposition additive non transitive pourrait convenir à notre situation, où, en raison de la rationalité limitée de l'utilisateur, nous nous trouvons dans un cas non transitif. Cependant, le nombre de solutions possibles est très important, et nous ne pouvons pas faire comme dans TMA 02 [TMA 02] ; c'est-à-dire, mettre en place des équations pour déterminer une solution en utilisant tous les précédents éléments et les satisfaire, ainsi que l'élément courant. Nous devons utiliser une autre méthode. C'est pour cela que nous avons développé deux méthodes pour effectuer la désagrégation de l'information, dont une utilise le principe de la décomposition additive non transitive.

Chapitre V. Méthodes de calcul

Afin de mettre en œuvre notre système, plusieurs méthodes de calculs sont développées. Ces méthodes peuvent être des méthodes classiques comme celle utilisée pour calculer un score, ou peuvent avoir été définies par nos soins comme les méthodes de désagrégation 1 et 2. En effet étant donné les caractéristiques du système que nous mettons en œuvre, il a été nécessaire de définir d'autres méthodes pour effectuer des opérations de désagrégation.

Mais commençons par le début, à savoir l'initialisation du système.

V.1. Initialisation du système

Lors de la création de leur profil, dans certains cas on demande aux utilisateurs de fournir quelques informations à propos de leurs préférences afin de pouvoir initialiser leur profil. Grâce à cette initialisation, on va pouvoir effectuer le tri initial. Dans le cas où l'on ne demande pas d'informations à l'utilisateur (investissement moindre de l'utilisateur), le tri initial se fait de manière aléatoire.

V.1.A. Initialisation sans informations de l'utilisateur

Le système que nous proposons permet de se passer d'une phase d'initialisation du profil de l'utilisateur afin de correspondre complètement à un système incrémental.

Dans ce cas-là, on initialise comme ceci :

$$s(0)_i = 0 \tag{24}$$

- $s(0)_i$ correspond au score du critère i lors de l'initialisation ($t=0$)

Ainsi on peut utiliser le système sans que l'utilisateur ne donne aucune information afin d'initialiser son profil. Cela permet au système d'être utilisé sur des plateformes sans

que l'utilisateur n'ait à intervenir pour expliciter directement ses préférences lors de l'initialisation de son profil. Lors de la première, et uniquement la première, interaction de l'utilisateur avec la plateforme, les éléments qui lui seront présentés le seront dans un ordre aléatoire.

V.1.B. Initialisation avec des informations de l'utilisateur

Dans le cas où l'on a la possibilité de demander, lors de son inscription, à l'utilisateur des informations sur ses préférences, on utilise un formulaire que l'utilisateur peut remplir de manière plus ou moins complète selon son envie (au minimum un critère), afin d'initialiser son profil.

L'utilisateur pourra entrer des préférences et mettre une note à ses préférences afin que son profil puisse être initialisé. Il peut être intéressant de placer un multiplicateur global par rapport à des poids présents uniquement à la création du profil. Cela permet à l'utilisateur de donner une préférence à la fois sur un critère très précis, comme pour un film, le genre du film comme « Aventure », ou une échelle de prix comme « 5€-10€ », ou dans le cas du poids, donner une préférence vis-à-vis du prix en général par rapport au genre.

Pour initialiser le profil de l'utilisateur, on définit de manière fixe plusieurs groupes de critères auxquels on associe pour chacun un poids p_j , et on utilise le calcul suivant :

$$s(0)_i = \frac{\gamma_i * p_j}{\sum_{i=1}^L \gamma_i} \quad (25)$$

- γ_i correspond au score du $i^{\text{ème}}$ critère, $n \geq 1, n \in N$ et $i \in [1, \dots, n]$
- p_j correspond au poids associé au $j^{\text{ème}}$ groupe avec $m \geq 1, m \in N$ et $j \in [1, \dots, m]$, et $\sum_{i=1}^m p_j = 1$
- L correspond au nombre de critères
- $s(0)_i$ correspond au score du critère i à $t=0$

Voici un exemple :

Soit un film défini par 3 critères, les critères 1 (action) et 2 (aventure) appartenant au groupe A (style) possédant un poids de 0.7, et le critère 3 (prix d'achat de 5€-10€) appartenant au groupe B (prix) possédant un poids de 0.3.

L'utilisateur définit le score des critères sur une échelle allant de 0 à 5. Les critères se voient respectivement attribuer les scores suivant : 2, 4, 5.

En appliquant la formule, on obtient :

$$s(0)_{action} = \frac{2 * 0.7}{11} = 0.127$$

$$s(0)_{aventure} = \frac{4 * 0.7}{11} = 0.25$$

$$s(0)_{prix \text{ d'achat de } 5\text{€}-10\text{€}} = \frac{5 * 0.3}{11} = 0.136$$

Cependant, ce type d'initialisation est très peu utilisé, car dans la plupart des cas on souhaite ne pas demander explicitement ses préférences à l'utilisateur. Ainsi pour la suite, on considère que le profil de l'utilisateur est initialisé avec la première méthode (sans informations)

Nous cherchons dans notre système à ordonner des objets en fonction du profil de l'utilisateur en sortie du processus, c'est pourquoi il nous faut une méthode afin de pouvoir ordonner les différentes solutions présentées à l'utilisateur en fonction de son profil.

V.2. Ordonner les éléments

Afin de pouvoir construire un ordre entre les éléments présentés à l'utilisateur, nous avons le choix entre deux méthodes. Premièrement, une méthode de calcul de score global représentant l'intérêt global de l'utilisateur pour un élément donné et une seconde méthode, basée sur l'utilisation d'un prédicat binaire.

La méthode utilisant un prédicat binaire peut poser des problèmes. En effet lors de la comparaison de deux éléments, des problèmes d'incomparabilités apparaissent. Lorsque l'on compare les critères entre eux, certains critères peuvent être considérés comme étant incomparables les uns avec les autres.

De plus, ce que l'on cherche à faire ici est plus lié à du tri (ranking) qu'à de la classification, l'on veut pouvoir présenter à l'utilisateur des éléments, et cela de manière

ordonnée en fonction de son profil. C'est pourquoi nous utilisons un calcul de score (on utilise les principes de la théorie Multiattribut), cela nous permet d'avoir une estimation du degré de préférence d'une alternative par rapport à une autre, ce que ne permet pas la méthode du prédicat binaire.

V.3. Calcul du score

Pour chaque objet on calcule un degré de similarité par rapport au profil, afin de déterminer à quel point cet objet plait à l'utilisateur, et ce d'après son profil. Ce score est utilisé pour sélectionner des objets lors de la recherche d'informations, trier la liste des objets que l'on présente à l'utilisateur (ordre de préférence décroissant).

Étant utilisé dans le cas d'un apprentissage par renforcement il est naturel que l'on utilise une fonction de type somme, car l'algorithme par renforcement utilise le principe de l'utilité.

On calcule le score de l'objet de la façon suivante :

$$\text{Score de l'élément } (t) = \frac{\sum_{i=1}^{i=\text{nombre de critères dans l'élément}} s(t)_i}{\text{nombre de critères dans l'élément}} \quad (26)$$

- $s(t)_i$ correspond au score du critère i au moment t dans le profil de l'utilisateur avec $s(0)_i = 0$

On remarquera que le score de l'élément est ainsi une somme du score de chacun des critères le composant. Pour chaque critère, son score est défini en fonction de l'importance du critère vis-à-vis du profil de l'utilisateur. Afin de ne pas privilégier les éléments composés de nombreux critères, on divise cette somme par le nombre de critères composant l'élément.

Cette méthode permet de se passer d'une matrice de corrélation des objets entre eux et de sa maintenance, ce qui représente un gain important en coût de calcul.

Elle permet également de proposer de manière plus directe, des objets possédant des caractéristiques qui sont présentes de manière séparée dans des éléments que l'utilisateur a aimés.

Par exemple :

Voici 2 livres aimés par l'utilisateur dans leur ordre de préférence.

Livre 1 = {Science, Informatique}

Livre 2 = {Poésie, Littérature}

Et voici d'autres livres disponibles :

Livre 3 = {Science, Informatique, JAVA, C++}

Livre 4 = {Poésie, Littérature, Romantisme}

Livre 5 = {Science, Poésie}

Un système de matrice de corrélation ordonnerait les livres à l'utilisateur comme ceci : {Livre 3, Livre 4, Livre 5}.

Car ce système de matrice de corrélation aurait proposé le livre étant le plus corrélé avec le livre préféré par l'utilisateur (livre 1) et ensuite le livre étant le plus corrélé avec le livre 2. L'utilisation directe de la matrice n'aurait pas placé en première position le livre 5, qui pourtant regroupe des critères que l'utilisateur aime, mais qui sont présents de manières séparées dans les livres précédemment aimés par l'utilisateur. Le système de scoring que nous avons mis en place permet de mieux prendre en compte ce genre de cas.

Par ailleurs, afin de réaliser la désagrégation des éléments dans le but de mettre à jour le profil de l'utilisateur, nous avons mis au point deux méthodes.

Le but principal de ces deux méthodes consiste à déterminer la valeur de sc_i . sc_i correspondant à un score calculé pour le critère i à partir de l'objet courant (sélectionné ou noté). Et cela, afin de faire varier la valeur de i dans le profil de l'utilisateur, et d'avoir un profil le plus représentatif des préférences de l'utilisateur.

En effet, **l'idée principale étant de désagréger le mieux possible les données dont on dispose afin d'avoir un score calculé sc_i correspondant au mieux aux préférences de l'utilisateur.**

V.4. Algorithme n°1 : Désagrégation à partir d'une note

Dans cette méthode, l'utilisateur fournit une note sur un objet. Cette donnée est considérée comme étant explicite, puisque fournie directement par l'utilisateur.

La note donnée par l'utilisateur est fonction de l'échelle mise en place, mais le minimum de cette échelle est toujours supérieur ou égal à 0. Ceci afin que les utilisateurs ne puissent pas donner de notes punitives. En effet, le système que l'on a mis en place se propose de ne faire que croître la valeur des critères contenus dans le profil de l'utilisateur. Dans l'absolu, le fait de ne mettre que des notes positives est équivalent au fait de mettre des notes négatives, car pour comparer deux critères entre eux, la différence entre les valeurs de chaque critère est toujours employée. De plus, les utilisateurs sont habitués à n'utiliser que des échelles de notes positives (systèmes des étoiles), c'est pourquoi nous n'utilisons que des notes supérieures ou égales à 0.

Un calcul de répartition des scores est effectué pour chaque attribut se basant sur la note fournie par l'utilisateur, et nous intégrons le vecteur de critères calculé résultant au profil de l'utilisateur.

Calcul de désagrégation en fonction de la note :

$$sc_i = \frac{\text{note}}{\text{nombre de critères de l'élément}} \quad (27)$$

- sc_i correspond au score calculé pour le critère i à partir de l'objet courant

Bien que cette représentation puisse être considérée comme naïve, elle nous permet d'utiliser ce système avec n'importe quel type d'objet. Pour ce faire, nous définissons le type de l'objet comme inconnu. Ainsi nous ne privilégions pas un critère par rapport à un autre, et dans notre représentation, tous les critères de l'objet ont un score équivalent.

Avec ces scores calculés, nous mettons à jour le profil de l'utilisateur. Pour ce faire, nous utilisons la formule suivante :

$$s(t+1)_i = s(t)_i + 0.1 * \log(1 + sc_i) \quad (28)$$

- $s(t+1)_i$ correspond au score du critère i au temps $t+1$ avec $s(0)_i = 0$

On utilise une fonction logarithmique afin de réduire l'impact des valeurs élevées. Un coefficient de 0.1 est utilisé afin de contrôler l'impact de la nouvelle donnée sur le système (obtenu par expérimentations dans [TMA 02], [BOUG et al. 04]). On rajoute 1 afin d'éviter les valeurs négatives obtenue avec le logarithme.

Afin de comprendre l'intérêt d'utiliser cette formule, nous effectuons la démonstration suivante :

$$\begin{aligned} s(t+1)_i &= s(t)_i + 0.1 * \log(1 + sc_i) \\ \Leftrightarrow s(t+1)_i - s(t)_i &= 0.1 * \log(1 + sc_i) \\ \Leftrightarrow \Delta_{s_i} &= 0.1 * \log(1 + sc_i) \end{aligned}$$

Avec Δ_{s_i} , la variation de la valeur du critère i dans le profil de l'utilisateur entre le moment t et $t+1$.

Ainsi l'on voit bien que la variation de la valeur du critère i pour l'utilisateur ne dépend que de sc_i . Ce qui nous permet de faire varier la valeur de i de manière très importante si nécessaire. Pour par exemple modifier le profil utilisateur afin de refléter un récent changement de préférences.

Nous avons mis au point une autre méthode permettant d'exploiter les informations données de manière implicites par l'utilisateur lors de sa navigation sur une plateforme effectuant des recommandations à l'utilisateur. Cette méthode permet de se passer d'informations explicites que l'utilisateur doit fournir (cf algorithme n°1, V.4).

V.5. Algorithme n°2 : Désagrégation à partir de paires d'objets (Pairwise)

Quand un utilisateur choisit un objet parmi la liste (triée en fonction des scores obtenus de chaque élément qu'on lui propose), on utilise un apprentissage mettant en œuvre un calcul de type « Pairwise comparison » (comparaison d'objets 2 à 2). À partir de ceux-ci,

on extrait des données implicites. Plusieurs avantages à ce type de méthodes sont décrits dans [FUR 10].

Afin de calculer le score de chaque critère, on utilise la décomposition additive non transitive. En effet, on ne peut pas affirmer que pour le décideur si $A > B$ et $B > C$ alors $A > C$ (avec A , B et C étant des objets multicritère), étant donné la nature des objets sur lesquels on travaille et du fait des utilisateurs ayant une rationalité limitée. On peut prendre un exemple simple. Si l'utilisateur doit choisir un dé pour un jeu de hasard, avec les dés suivants :

Soit trois dés à 6 faces A , B , C suivant :

- le dé A porte les numéros $\{2, 2, 4, 4, 9, 9\}$ sur ses faces ;
- le dé B porte $\{1, 1, 6, 6, 8, 8\}$;
- le dé C porte $\{3, 3, 5, 5, 7, 7\}$.

La probabilité que A donne un plus grand nombre que B est $5/9$ ($20/36$) ; et donc pour l'utilisateur $A > B$.

La probabilité que B donne un plus grand nombre que C est $5/9$; et donc pour l'utilisateur $B > C$.

La probabilité que C donne un plus grand nombre que A est $5/9$; et donc pour l'utilisateur $C > A$, or cela est impossible avec un modèle transitif. C'est pour cela que nous utilisons une décomposition additive non transitive.

Pour ce faire, nous utilisons la procédure suivante, nous construisons un vecteur valué et nous l'utilisons pour la mise à jour du profil de l'utilisateur. Pour cela, on effectue plusieurs calculs intermédiaires, qui viennent mettre à jour le vecteur temporaire au fur et à mesure que les interactions se déroulent. Ces interactions correspondent à des couples d'objets constitués à chaque fois de l'objet sélectionné par l'utilisateur et d'objet proposés à l'utilisateur et qui sont situés avec un index plus faible que celui que l'utilisateur a sélectionné ; qui sont utilisés tour à tour afin de mettre à jour un vecteur temporaire.

Nous travaillons avec l'objet sélectionné par l'utilisateur et initialisons le vecteur temporaire avec un vecteur obtenu à partir d'un calcul effectué avec l'algorithme par note (formule n°27). Les calculs utilisent également les éléments affichés à l'utilisateur, qui se situent au-dessus de l'élément sélectionné. Au-dessus correspond, dans ce travail, à la position d'un élément à un index plus faible, dans la liste présentée à l'utilisateur, que celui sélectionné. En effet, plus on est situé bas dans la liste et plus l'index est grand, et plus le

score calculé est faible, et inversement, plus on est situé haut dans la liste et plus l'index est faible, et plus le score calculé est élevé.

Nous effectuons les calculs en utilisant l'objet sélectionné (préféré) par l'utilisateur et tour à tour les objets au-dessus de celui-ci, en faisant des calculs par paires. Ce principe reflète l'idée que dans une liste triée, lorsque l'utilisateur sélectionne un élément qui n'est pas en première position, cela correspond au fait que l'utilisateur considère l'élément sélectionné comme étant préféré aux précédents [JOA 02]. Nous n'utilisons pas les éléments situés en dessous de l'élément sélectionné, car on ne peut pas affirmer que l'utilisateur en a pris connaissance et donc qu'il a préféré l'élément sélectionné à ceux-ci.

Nous mettons ainsi à jour le vecteur temporaire.

Voici le processus principal :

- 1) Initialiser un vecteur temporaire en utilisant un calcul de désagrégation en fonction de la note (algorithme n°1) avec l'objet sélectionné par l'utilisateur, et une note moyenne équivalente à $note_max/2$ ($note_max$ dépendant de ce qui est défini comme échelle note pour le système).
- 2) Retrait des critères non discriminants (critère en commun avec les deux objets) entre l'objet sélectionné par l'utilisateur et un objet au-dessus de celui-ci. On utilise le principe d'indépendance préférentielle (ou "séparabilité") : la préférence entre un objet a et un objet b ne dépend pas des attributs en commun de a et b. En effet, la préférence globale entre deux alternatives ne dépend pas des attributs qui ne les départagent pas.
- 3) Faire un calcul de répartition, et mettre à jour le vecteur temporaire avec les données d'obtenues.
- 4) Nous répétons la tâche 1 à 3, avec un autre objet au-dessus de celui sélectionné, jusqu'à ce qu'il n'y ait plus aucun autre objet au-dessus de celui sélectionné par l'utilisateur, non traité.
- 5) Mise à jour du profil de l'utilisateur en utilisant le vecteur temporaire ainsi obtenu après les divers calculs opérés pour chaque paire d'objets.

Dans le but de faire le calcul de répartition, nous utilisons une décomposition additive non transitive par différence. Voici le fonctionnement que nous avons mis en place :

$s(t)_{ij}$ et $s(t)_{ik}$ correspondent respectivement au score du critère i au moment t dans le profil de l'utilisateur pour un objet j et k . Avec k étant l'objet sélectionné par l'utilisateur, et j étant un objet situé au-dessus de celui sélectionné.

Nous avons :

$$\frac{\sum_{i=1}^{i=\text{nombre de critères}} s(t)_{ij}}{\text{nombre de critères de l'élément}} > \frac{\sum_{i=1}^{i=\text{nombre de critères}} s(t)_{ik}}{\text{nombre de critères de l'élément}} \quad (29)$$

En effet, car l'objet j est plus haut dans la liste (triée) présentée à l'utilisateur, que l'objet k .

Cependant, nous voulons :

$$\frac{\sum_{i=1}^{i=\text{nombre de critères}} s(t)_{ij}}{\text{nombre de critères de l'élément}} < \frac{\sum_{i=1}^{i=\text{nombre de critères}} s(t)_{ik}}{\text{nombre de critères de l'élément}} \quad (30)$$

Cela montrerait que le profil de l'utilisateur que l'on a construit est correct.

Pour corriger le profil de nos utilisateurs, nous devons déterminer quels critères et avec quel score nous allons mettre à jour leur profil.

Pour cela nous prenons la différence de score entre nos deux objets.

$$\Delta = \frac{\sum_{i=1}^{i=\text{nombre de critères}} s(t)_{ij}}{\text{nombre de critères de l'élément}} - \frac{\sum_{i=1}^{i=\text{nombre de critères}} s(t)_{ik}}{\text{nombre de critères de l'élément}} \quad (31)$$

En effet, nous avons deux objets avec leur score calculé. Nous voulons que le score de l'objet sélectionné soit supérieur au score de l'objet au-dessus.

Pour ce faire, nous prenons la différence de score (Δ) et nous l'utilisons pour la mise à jour du profil de l'utilisateur. Δ est toujours positif, car le score de l'objet j est toujours

supérieur ou égal à celui de k , cela étant dû au fait que ces objets sont ordonnés de manière décroissante en fonction du score calculé de chaque objet.

Nous utilisons ainsi Δ afin d'obtenir le score calculé de chaque critère composant notre vecteur temporaire. Pour cette méthode (pairwise) nous redéfinissons sc_i ainsi :

$$sc_i = \frac{\Delta}{\text{nombre de critères de l'élément}} \quad (32)$$

Ainsi plus la différence de score est importante, et plus l'on peut en déduire que le système n'est pas bien réglé. On voit également que plus la différence de score entre le premier élément de la liste et l'élément sélectionné par l'utilisateur est grande et plus il faut y remédier. Et c'est ce qui est fait grâce à Δ , car plus la différence est grande et plus Δ est grand et donc plus sc_i a une valeur importante. Cela nous permet de corriger rapidement le système. En effet, lorsque nous avons un Δ ayant une valeur importante, nous avons un grand impact sur le profil de l'utilisateur. Ce qui nous permet d'augmenter la vitesse d'adaptation de notre système par rapport à des changements de préférence de l'utilisateur.

Donc, nous avons un score calculé pour chaque critère à l'intérieur du vecteur valué. Nous pouvons utiliser ce vecteur pour mettre à niveau le profil de l'utilisateur.

Nous utilisons ici une formule semblable à celle utilisée dans le calcul de la note pour la mise à jour du profil de l'utilisateur (formule n°28). Nous ajoutons une variable qui va nous permettre d'avoir un impact équilibré sur le système. En effet, afin de maintenir un système efficace, nous voulons un impact équilibré sur le système et pour cela nous avons besoin de prendre en compte le nombre de calculs par paires que nous avons faits.

Par conséquent, nous utilisons la formule suivante :

$$s(t+1)_i = s(t)_i + 0.1 * \log(1 + \alpha * sc_i) \quad (33)$$

Avec :

$$\alpha = \frac{1}{\text{nombre d'éléments au-dessus de celui sélectionné}} \quad (34)$$

- $s(t + 1)_i$ correspond au score du critère i au temps $t+1$
- $s(t)_i$ correspond au score du critère i au temps t , c'est-à-dire que $s(t + 1)_i$ est le nouveau score du critère i et $s(t)_i$ est l'ancien score du critère i .
- sc_i correspond au score calculé pour le critère i à partir de l'objet courant

Nous avons ainsi deux façons de mettre à jour le profil de l'utilisateur, en fonction des informations explicites (note) et implicites (sélection par l'utilisateur).

V.6. Conclusion

Le système proposé ici met en œuvre différentes techniques afin de réaliser l'apprentissage du profil des utilisateurs et des recommandations qui exploitent ces profils.

Ces techniques permettent de prendre en compte les données explicites (notes) et implicites (sélection) induites par les actions des utilisateurs. De traiter ces données, afin de mettre à jour de manière continue les profils des utilisateurs, et de faire des recommandations aux utilisateurs en triant les solutions qui leur sont présentées en fonction de ses préférences.

Il est à noter que, comme souhaité, la valeur des critères au sein des profils utilisateurs ne fait que croître, et ce sans borne supérieure.

Les techniques mises en œuvres ici permettent d'apporter en parties des solutions à des verrous technologiques comme la désagrégation de critères et la prise en compte d'un nombre variable de critères dans le processus d'aide à la décision interactif, et ce sans définir au préalable de famille cohérente de critères sur laquelle est basée la décision.

Chapitre VI. Utilisation des dépendances/interactions entre les critères

Dans le système que nous avons mis en place, les critères sont considérés comme indépendants. Or la plupart des critères ne sont pas indépendants les uns des autres. Par exemple, la puissance d'une voiture dépend de son moteur. Il y a donc des liens entre les critères. Ces liens peuvent être utilisés pour améliorer la désagrégation et/ou l'agrégation des critères.

Nous nous sommes principalement intéressés à l'utilisation de méthodes permettant d'exploiter les liens entre les critères afin d'améliorer les capacités de désagrégation de notre système.

Contrairement aux moyennes arithmétiques pondérées, les intégrales floues sont capables de prendre en considération les interactions éventuelles parmi les attributs ou critères. Il existe plusieurs classes d'intégrales floues, parmi lesquelles les plus représentatives sont celles de Choquet et Sugeno [BOUY et al. 06].

Afin d'agréger des valeurs définies sur une échelle d'intervalle, les intégrales de Choquet sont appropriées. Or dans ce travail nous travaillons sur des valeurs définies sur une échelle d'intervalle (par exemple les prix sont définis sur une échelle d'intervalle).

De plus, nous voulons prendre en compte des phénomènes d'interaction entre critères, or il a été suggéré d'utiliser une intégrale de Choquet [GRA 04] afin de permettre cette prise en compte, ce qui renforce notre intérêt pour cette méthode.

On remarquera également que "L'intégrale de Choquet peut être vue comme le moyen le plus simple pour étendre à des alternatives quelconques le raisonnement d'un décideur sur des alternatives binaires." [KOJ 06]. Or notre algorithme Pairwise, correspond à un raisonnement sur des alternatives binaires (choix d'un objet par rapport à un autre), cette méthode permettra ainsi, grâce aux intégrales de Choquet d'étendre ce raisonnement, afin de l'utiliser pour effectuer le tri (ranking) des solutions les unes par rapport aux autres.

C'est pour les raisons évoquées ci-dessus que l'on va mettre en œuvre des intégrales de Choquet au sein de notre système.

Afin de prendre en compte les interactions entre les critères et attributs, nous avons ainsi mis en œuvre des intégrales de Choquet. Nous utilisons ces Intégrales afin de déterminer l'importance de chaque critère dans le choix de l'utilisateur en fonction de ses actions et en prenant en compte les interactions entre les critères. En effet, nous pouvons utiliser ces interactions afin d'améliorer la désagrégation des critères d'un objet. Nous

obtenons des informations sur les interactions des critères entre eux en comparant les préférences que l'utilisateur expose à travers ses actions. Lors de cette comparaison, il est possible de déterminer quels sont les critères qui interagissent entre eux et si cette interaction est positive ou négative.

VI.1. Mise en œuvre

Afin de pouvoir exploiter de manière simple des intégrales de Choquet, un package nommé « Kappalab » a été mis à la disposition des utilisateurs à travers le système statistique R [MEY et al. 06]. Ce package contient des routines pour la manipulation de divers types de fonctions tels que des capacités. Il peut être utilisé pour calculer plusieurs intégrales non-additives : l'intégrale de Choquet, l'intégrale de Sugeno. Une analyse des capacités, en termes de comportement de décision, peut être réalisée par le calcul des différents indices [voir chapitre III.3.C] tel que la valeur de Shapley, l'indice d'interaction, etc. La célèbre transformation de Möbius, ainsi que d'autres représentations équivalentes de fonctions peuvent également être calculées.

Nous avons transformé nos données afin de pouvoir les utiliser au sein de ce package. En effet, les données que nous mettons en œuvre ne sont pas utilisables dans leur état initial par Kappalab, il nous a donc fallu les convertir dans un format exploitable par Kappalab.

Pour ce faire à chaque appel à ce package, nous listons tous les critères faisant partie des éléments à évaluer et convertissons ces informations. C'est-à-dire que pour un élément (K) parmi une liste (A, B, K, C), nous allons convertir les informations à propos de ces éléments en variables booléennes. Pour ce faire, nous listons tous les critères composant les éléments, et pour chaque élément on construit un tableau à deux dimensions contenant le nom de ces critères et, pour chaque critère composant les éléments, nous définissons la valeur du critère par 0 si le critère ne compose pas l'élément et 1 s'il le compose.

Il faut à présent déterminer les contraintes à insérer dans le système afin que l'on puisse utiliser les intégrales de Choquet pour déterminer une solution et les utilités de chaque critère et paire de critères.

VI.1.A. Déterminer les contraintes

Les contraintes sont sous la forme d'un ordre faible (weak order) entre différents éléments. Ensuite, nous intégrons ces informations dans kappalab qui constituent chacune une contrainte.

Afin de déterminer les contraintes, nous avons deux possibilités :

Le cas où une note est fournie et le cas où l'utilisateur choisit un élément dans la liste qui lui est proposée.

Dans le premier cas où l'utilisateur a fourni une note, nous pouvons, de manière naturelle, ordonner l'élément que l'utilisateur vient de noter avec les autres éléments qu'il a noté précédemment et utiliser ces données.

Par exemple, nous avons :

Élément	Note
A	17
K	15
B	14
C	14

Tableau 4. *Exemple d'association d'éléments et d'une note*

Dans ce cas-là (Tableau 4), l'on obtient comme contraintes :

- $A > K$
- $A > B$
- $A > C$
- $K > B$
- $K > C$

Dans le second cas où l'utilisateur a choisi un élément parmi d'autres (K a été choisi parmi A, B, C et K), nous obtenons l'information que l'élément sélectionné est préféré aux autres par l'utilisateur. Nous considérons ici que l'utilisateur a pris connaissance de tous les éléments présents, et qu'ainsi sa sélection reflète bien qu'il préfère l'élément sélectionné par rapport à tous les autres.

Dans ce cas-là, l'on obtient comme contraintes :

- $K > A$
- $K > B$
- $K > C$

Ces contraintes sont utilisées les unes après les autres. Ce sont ces contraintes qui vont servir de base afin de déterminer une solution qui va satisfaire chacune de ces contraintes.

En effet, c'est grâce à cette solution que nous allons pouvoir effectuer des calculs (intégrales de Choquet, représentation de Möbius), afin de déterminer les utilités de chaque critère et des paires de critères.

VI.1.B.Calcul des utilités

En pratique, afin de diminuer le nombre d'inconnues, des classes spéciales d'intégrales de Choquet sont généralement utilisées ; elles sont obtenues en considérant des capacités k -additives. Pour notre calcul nous allons utiliser une capacité 2-additive, car pour de gros problèmes, tant pour des raisons de calcul que de simplicité, il peut être préférable de limiter l'ensemble des solutions possibles pour des capacités k -additives, avec $k \in \{1, \dots, n\}$, $n \geq 1$ et $n \in \mathbb{N}$, l'on utilise généralement $k = 2$ ou 3 . On utilise donc une capacité 2-additive afin d'obtenir une solution prenant en compte les liens entre les éléments 2 à 2.

Il nous faut à présent trouver une solution qui puisse être en accord avec toutes ces contraintes.

Pour ce faire 3 approches existent :

- L'approche des moindres carrés [MOR 89] : cette approche minimise la distance quadratique moyenne entre les utilités globales des éléments.
- Une approche basée sur la programmation linéaire [MAR 00b] : Maximiser la plus petite différence entre les utilités globales des éléments dans un ordre de priorité défini.
- L'approche de la variance et de la distance minimum [KOJ 07] : Favoriser la capacité « la moins spécifique » compatible avec les préférences initiales de l'utilisateur (les contraintes définies précédemment).

Nous avons choisi d'utiliser l'approche de la variance et de la distance minimum afin de favoriser la capacité « la moins spécifique ».

Cela nous permet d'obtenir une solution avec une capacité 2-additive sous la forme de sa représentation de Möbius. Plus d'informations à ce propos peuvent être trouvées dans [GRA et al. 08]. Les valeurs d'utilité sont comprises entre -1 et 1.

En voici un exemple de représentation sous la forme de Möbius pour 5 critères :

Critères et Paires de critères	Utilité des critères
{}	0.00
{1}	0.46
{2}	0.30
...	...
{3, 5}	0.06
{4, 5}	-0.07

Tableau 5. *Exemple de représentation sous la forme de Möbius pour 5 critères*

On obtient ainsi l'utilité de chaque critère, par exemple : {0}, {1} ; et également des paires de critères par exemple : {3, 5}, qui correspond à l'utilité de la paire des critères 3 et 5. Ainsi, les interactions, positives ({3, 5} avec 0.06), ou négatives ({4, 5} avec -0.07) sont prises en compte.

Grâce à cela nous pouvons donc prendre en compte les interactions des critères les uns avec les autres.

VI.1.C. Utilisation des données à l'issue des intégrales de Choquet

Afin d'utiliser ces données à propos de la valeur d'utilité des critères ou des paires de critères, nous avons défini plusieurs méthodes faisant intervenir les intégrales de Choquet, et plus spécifiquement une capacité 2-additive sous la forme de sa représentation de Möbius.

VI.1.C.i. Méthode directe

La première méthode correspond au fait d'exploiter directement ces données, et de les intégrer directement dans le profil de l'utilisateur en utilisant la formule suivante :

$$s(t+1)_i = s(t)_i + 0.1 * \log(1 + uc_i) \quad (35)$$

Avec

$$uc_i = \{i\} + \sum \{i, x\} \quad (36)$$

- $s(t+1)_i$ correspond au score du critère i au temps $t+1$ avec $s(0)_i = 0$
- $s(t)_i$ correspond au score du critère i au temps t
- uc_i correspond à l'utilité calculée du critère i obtenu précédemment en utilisant la représentation de Möbius du critère i et des paires de critères faisant intervenir i
- x correspond à un critère faisant partie d'une paire où i intervient

La valeur d' uc_i prend ainsi simplement la place de sc_i que l'on calculait grâce aux algorithmes de désagrégation 1 et 2. Et un paramètre égal à 1 est ajouté au logarithme afin de ne pas ajouter de valeurs négatives, en effet, en fonction des synergies entre les critères, s'il y a trop de synergies négatives faisant intervenir i et si $\{i\}$ a une valeur trop faible (par rapport à la somme des synergies), alors la valeur d' uc_i peut être négative.

Cette méthode permet donc de complètement remplacer les algorithmes de désagrégation que nous avons mis en place, par les intégrales de Choquet. L'avantage étant qu'avec cette méthode, les interactions entre les critères sont prises en compte. Par contre, les performances du système sont entièrement limitées par les performances des intégrales de Choquet.

Cette méthode peut être utilisée afin de comparer les performances qu'offrent les intégrales de Choquet. Elle peut également servir de comparaison avec les performances de nos algorithmes et également lorsque l'on utilise la méthode décrite ci-après qui permet d'utiliser les intégrales de Choquet en complément de nos systèmes de désagrégation.

VI.1.C.ii. Méthode d'utilisation en tant que facteur multiplicateur

La deuxième méthode correspond à l'idée que nos algorithmes de désagréations sont efficaces, mais qu'ils nécessitent un complément afin d'être encore plus efficaces, ce qui est le cas puisqu'ils ne prennent pas en compte les liens/interactions entre les critères qui vont permettre de mieux considérer une grande quantité de critères. Ainsi, l'idée est d'utiliser les données obtenues sous la forme de Möbius en tant que facteur multiplicateur sur le score à rajouter au score actuel du critère i .

En effet, les valeurs d'utilité issues de la capacité 2-additive sous la forme de sa représentation de Möbius, sont comprises entre -1 et 1, ce qui se prête particulièrement à être utilisé en tant que facteur multiplicateur.

Nous utilisons ainsi la formule suivante :

$$s(t+1)_i = s(t)_i + uc_i * 0.1 * \log(1 + sc_i) \quad (37)$$

Avec

$$uc_i = \{i\} + \sum \{i, x\} \quad (38)$$

- $s(t+1)_i$ correspond au score du critère i au temps $t+1$ avec $s(0)_i = 0$
- $s(t)_i$ correspond au score du critère i au temps t
- uc_i correspond à l'utilité calculée du critère i obtenu précédemment en utilisant la représentation de Möbius du critère i et des paires de critères faisant intervenir i
- x correspond à un critère faisant partie d'une paire où i intervient
- sc_i correspond au score calculé pour le critère i grâce à la méthode Pairwise

Cette méthode permet ainsi d'utiliser à la fois les intégrales de Choquet, et également nos algorithmes de désagréations, les deux méthodes venant se compléter. Cette méthode à l'avantage de permettre d'augmenter ou de réduire l'impact sur le profil utilisateur de la méthode de mise à jour de profil, et ce en fonction de l'interaction du critère avec les autres critères. Et grâce à l'impact de l'interaction du critère, nous pouvons ainsi mieux prendre en compte de nombreux critères.

On remarque que comme précédemment la valeur d' uc_i peut être négative. Or dans le cas présent, où uc_i est utilisé en tant que facteur multiplicateur, la valeur de $s(t + 1)_i - s(t)_i$ peut être négative. L'idée, au sein de notre système, de faire évoluer de manière négative la valeur d'un critère dans le profil d'un utilisateur est contraire à notre souhait exprimé au chapitre IV.2.B. Cependant, il pourrait être intéressant de voir si effectivement, dans notre système, le fait de faire évoluer négativement la valeur d'un critère permettrait d'améliorer les performances du système. C'est une perspective que nous n'avons pas encore explorée.

VI.2. Conclusion

Les intégrales de Choquet nous permettent ainsi de prendre en compte les liens, interactions entre les critères. Les données extraites de ces interactions sont utilisées au sein de plusieurs méthodes dans le but de mettre à jour le profil de l'utilisateur, et ce, en prenant en compte ces liens. Ainsi le profil de l'utilisateur est plus cohérent et nous prenons mieux en compte la présence et les interactions de nombreux critères.

Une de nos perspectives consiste à tester si lors de l'utilisation des intégrales de Choquet, avec des valeurs uc_i négatives, cela mène à obtenir de meilleures performances.

Cependant, pour des raisons techniques, nous n'avons pas pu tester nos méthodes faisant intervenir les intégrales de Choquet. En effet, les méthodes disponibles au sein de Kappalab, présentent des limites au nombre de critères/attributs qui peuvent être utilisés. C'est en particulier la méthode implémentée afin d'obtenir une solution avec une capacité 2-additive qui pose problème. En effet, lorsqu'il y a trop de critères/attributs pris en compte (environ supérieur à 12, et nous utilisons très fréquemment plus de 12 critères/attributs), le système explose en consommation mémoire, et ne peut alors fournir de solution.

Une de nos perspectives consiste donc à trouver et à utiliser un autre moyen que Kappalab afin d'utiliser les intégrales de Choquet, et ce, afin de pouvoir tester l'impact que ses intégrales peuvent avoir sur les performances de notre système.

Il est également envisageable d'utiliser une alternative aux intégrales de Choquet, en utilisant les CP-nets avec les réseaux GAI [BOUT et al. 04], [QUE et al. 07].

Conclusion générale des contributions

Nous avons mis en place un système de profils multiattribut évolutifs, par apprentissage automatique et adaptatif et cela au sein d'un système de recommandation utilisé pour effectuer de l'aide à la décision.

Nous avons proposé un modèle apportant une solution cohérente à chaque contrainte qui a été définie.

- Ainsi nous avons mis au point un système qui est évolutif, c'est-à-dire que le ou les profils des utilisateurs évoluera/ont dans le temps, et l'assistance fournie évoluera de la même manière.
- Il met en œuvre les principes issus des domaines multicritère/multiattribut et cela en faisant varier dynamiquement les critères/attributs pris en compte pour la décision.
- Le système proposé est totalement incrémental, c'est-à-dire qu'il ne dispose pas de données *a priori* sur les utilisateurs, et de plus, il est synchrone.
- Il propose des recommandations génériques, c'est-à-dire qu'il peut être utilisé avec différents types d'éléments (une voiture, une maison, un film) et cela de manière naturelle.

Le système mis en place permet de mettre en œuvre un système supportant l'utilisation dynamique d'un nombre variable de critères pour effectuer de l'assistance aux décisions de l'utilisateur.

Il a été mis en place sur une plateforme en ligne (*mobifiz.fr*), et les temps de calcul sont transparents pour l'utilisateur (<1 sec).

Il permet d'utiliser des profils évolutifs qui offrent aux utilisateurs la possibilité de changer de préférences à n'importe quel moment et d'ainsi avoir un système qui s'ajuste de manière très rapide aux nouvelles préférences. Pour cela, nous avons également défini plusieurs méthodes permettant de réaliser de la désagrégation à base d'informations explicites et implicites, et cela dans le cas où les critères sont considérés indépendants les uns des autres, mais aussi dans le cas où des interactions entre les critères sont présentes.

À partir de ce travail, plusieurs perspectives ont été définies :

Une première perspective serait de déterminer de manière automatique, quel profil de l'utilisateur il est nécessaire d'utiliser, et ce en fonction du contexte dans lequel l'utilisateur se trouve.

Une seconde perspective serait d'utiliser des données contenues dans un profil au sein d'un autre, une sorte de système de profils collaboratif pour un même utilisateur. Cela permettrait d'obtenir des informations sur un profil que l'utilisateur n'a pas directement indiquées dans le contexte où il se trouve, mais où on peut retrouver l'information à partir d'un profil présentant des similarités.

Une troisième perspective consiste à tester si lors de l'utilisation des intégrales de Choquet, avec des valeurs uc_i négatives, cela mène à obtenir de meilleures performances. C'est-à-dire, si la possibilité, avec les intégrales de Choquet de pouvoir réduire la valeur d'un critère dans le profil de l'utilisateur, peut apporter de meilleures performances.

Une quatrième perspective consiste à trouver et à utiliser un autre moyen que Kappalab afin d'utiliser les intégrales de Choquet, et ce, afin de pouvoir tester l'impact que ces intégrales peuvent avoir sur les performances de notre système.

Une cinquième perspective serait d'utiliser une alternative aux intégrales de Choquet, en utilisant les CP-nets avec les réseaux GAI.

Une sixième perspective à notre travail serait d'utiliser les scores présents dans les profils des utilisateurs afin de déterminer quels sont les vrais critères, quasi-critère et pseudo-critère. Cela permettant ainsi d'offrir une visualisation des préférences de l'utilisateur.

Partie III. Expérimentations

Nous avons ainsi explicité le cadre dans lequel nous nous trouvons et détaillé le système que nous avons mis en place, ainsi que les méthodes associées. Il faut à présent tester les performances de notre système. Pour cela, nous commençons par détailler quelques méthodes permettant de comparer des systèmes de recommandations. Ensuite, nous expliquons quels tests nous avons mis en place, pourquoi et comment, ainsi que les résultats obtenus et l'analyse de ces résultats.

Chapitre VII. Méthodologie

Les systèmes de recommandation sont maintenant populaires à la fois commercialement et dans la communauté de la recherche, où de nombreuses approches ont été proposées pour fournir des recommandations. Afin de pouvoir comparer ces différents systèmes, il faut mettre en place diverses méthodes qui vont permettre de poser une base à partir de laquelle on va pouvoir comparer les différents systèmes de recommandation.

VII.1. Outils disponibles

Afin d'effectuer des tests permettant d'obtenir un indice de performance correspondant à la capacité qu'a un système à produire de bonnes recommandations nous avons recherché des métriques. De bonnes recommandations permettent de satisfaire les utilisateurs en plaçant dans les premières places de listes qui leur sont présentées, des éléments qui leur plaisent.

Voici un aperçu des différentes méthodes qui existent afin d'obtenir un indice permettant de représenter la capacité qu'a un système à faire de bonnes recommandations.

Ces méthodes peuvent être rangées en 3 catégories. La première contient les méthodes pour mesurer la précision d'un système effectuant de la prédiction de note. La seconde contient les méthodes de mesure de prévision d'utilisation. La troisième et dernière contient les méthodes permettant d'évaluer les systèmes mettant en œuvre des tris des solutions/objets proposés. Ces méthodes sont tirées de [SHAN 11].

VII.1.A. Mesures de précision de la prédiction de note

Dans certaines applications, comme dans le service de location de film de Netflix (*netflix.com*), le système de recommandation tente de prédire la note qu'un utilisateur donnerait à un élément (par de 1 à 5 étoiles). Dans de tels cas, afin de mesurer la performance du système, il est nécessaire de mesurer la précision des prédictions.

L'écart quadratique moyen, ou Root Mean Squared Error (RMSE) est peut-être la métrique la plus couramment utilisée pour évaluer la précision des notes prédites. Le système génère des notes prédites NP_{uo} pour un ensemble de tests T pour des paires utilisateur-objet (u, o) pour lesquels les vraies notes NV_{uo} sont connues.

En règle générale, les NV_{uo} sont connues bien qu'elles soient cachées lors d'une expérience hors ligne, ou parce qu'elles ont été obtenues grâce à une étude sur les utilisateurs ou une expérience en ligne.

L'écart quadratique moyen entre la note prédite et la note réelle est donné par la formule suivante :

$$RMSE = \sqrt{\frac{1}{|T|} * \sum_{(u,o) \in T} (NP_{uo} - NV_{uo})^2} \quad (39)$$

Une alternative très utilisée à RMSE est la mesure de l'erreur absolue moyenne, ou Mean Absolute Error (MAE).

$$MAE = \sqrt{\frac{1}{|T|} * \sum_{(u,o) \in T} |NP_{uo} - NV_{uo}|} \quad (40)$$

Par rapport à MAE, RMSE pénalise de manière disproportionnée des erreurs importantes, de sorte que, étant donné un ensemble de tests avec quatre valeurs de RMSE d'objets cachés, la méthode RMSE produirait un meilleur résultat avec un système qui fait une erreur de 2 (RMSE=2) sur trois notes et 0 sur la quatrième, à celui qui fait une erreur de 3

(RMSE=3) sur une note et 0 sur tous les trois autres, tandis MAE ne préférerait que le second système.

VII.1.B. Mesure de prévision d'utilisation

Dans de nombreux systèmes de recommandation, le système ne permet pas de prédire les préférences de l'utilisateur à propos d'éléments à lui proposer, tel que ce qui est fait lors de la prédiction de notes pour des films, mais essaie de recommander aux utilisateurs des éléments qu'ils pourraient utiliser (acheter), ou plus classiquement, qu'ils pourraient trouver intéressant (pertinent). Un exemple majeur de ces systèmes correspond aux moteurs de recherche.

Par exemple dans Netflix, lorsque des films sont ajoutés dans la liste personnelle des films de l'utilisateur (sa collection), le système propose un ensemble de films pouvant être intéressant, étant donné le film ajouté. Dans ce cas, nous ne nous intéressons pas à savoir si le système prévoit correctement les notations de ces films, mais plutôt de savoir si le système prévoit bien que l'utilisateur va ajouter ces films à sa collection (acheter le film).

Afin d'évaluer les prévisions d'utilisation, nous possédons généralement un ensemble de données constituées des éléments que chaque utilisateur a utilisé (ajouté à sa collection par exemple), et qui sont donc considérés comme étant pertinents pour l'utilisateur. Ensuite, nous sélectionnons un utilisateur, nous masquons une partie de ses sélections, et nous utilisons le système de recommandation afin qu'il prévoie un ensemble d'éléments que l'utilisateur va utiliser. Nous obtenons alors quatre résultats possibles pour les articles recommandés et cachés. Ces résultats sont présentés dans le tableau suivant (Tableau 6) :

	Élément recommandé	Élément non recommandé
Élément pertinent	Vrai positif (vp)	Faux négatif (fn)
Élément non pertinent	Faux positif (fp)	Vrai négatif (vn)

Tableau 6. *Classification des résultats possibles d'une recommandation d'un élément pour l'utilisateur*

Dans le cas où les tests se font de manière hors ligne (pas d'utilisation directe du système), puisque les données ne sont généralement pas recueillies par le système de

recommandation au cours de l'évaluation, nous sommes obligés de supposer que les éléments non choisis ne l'auraient pas été, et ce même s'ils avaient été recommandés, c'est-à-dire qu'ils sont sans intérêt ou inutiles à l'utilisateur. Cette hypothèse peut être fausse, comme lorsque l'ensemble des articles non choisis contient quelques articles intéressants que l'utilisateur n'a pas choisis. Par exemple, un utilisateur peut ne pas avoir choisi un élément parce qu'il n'était pas au courant de son existence, mais après que le système de recommandation lui ait proposé cet élément, l'utilisateur peut décider de le sélectionner. Dans ce cas-là, le nombre de faux positifs est surestimé.

Nous pouvons ainsi compter le nombre d'éléments de chaque case du tableau et calculer les quantités suivantes :

$$Precision = \frac{\text{nombre de } vp}{\text{nombre de } vp + \text{nombre de } fp}$$

(41)

La précision correspond au nombre d'éléments pertinents (vp) rapporté au nombre d'éléments total proposé par le système.

$$Rappel = \frac{\text{nombre de } vp}{\text{nombre de } vp + \text{nombre de } fn}$$

(42)

Le rappel est défini par le nombre d'éléments pertinents par rapport au nombre d'éléments pertinents total.

$$Taux \text{ de vrai positif} = \frac{\text{nombre de } fp}{\text{nombre de } fp + \text{nombre de } vn}$$

(43)

Le taux de vrai positif est défini par le nombre de faux positifs par rapport au nombre d'éléments non pertinents.

$$F - \text{mesure} = 2 \cdot \frac{\text{precision} \cdot \text{rappel}}{\text{precision} + \text{rappel}}$$

(44)

Cette mesure allie précision et rappel. C'est la moyenne harmonique de la précision et du rappel, la traditionnelle F-mesure.

Un système de recommandation parfait fournira des réponses dont la précision et le rappel sont égaux à 1. Dans ce cas-là, l'algorithme trouve la totalité des éléments pertinents (rappel), et il ne fait aucune erreur (précision). Dans la réalité, les systèmes sont plus ou moins précis, et plus ou moins pertinents. Il sera possible d'obtenir un système très précis (par exemple un score de précision de 0,99), mais peu performant (par exemple avec un rappel de 0.10, qui signifiera qu'il n'a trouvé que 10% des réponses possibles). Dans le même ordre d'idée, un algorithme dont le rappel est fort (par exemple 0.99 soit la quasi-totalité des éléments pertinents), mais la précision faible (par exemple 0.10) fournira en guise de réponse de nombreux éléments non pertinents en plus de ceux pertinents : il sera donc difficilement exploitable.

VII.1.C. Mesure d'ordre (ranking mesure)

Afin de mesurer l'ordre défini par un système de recommandation par rapport à un ordre de référence (bon ordre), plusieurs méthodes sont utilisées. Dans les cas où les notations (données explicites) sur les éléments, fournies par les utilisateurs, sont disponibles, nous pouvons classer les articles les mieux notés dans l'ordre décroissant des notes, avec des éléments liés (en anglais, ces éléments sont dits « *tie* »). Il est important de noter que ces liaisons n'ont rien avoir avec les liens/dépendance/interactions entre des attributs, vues au chapitre III.3 et VI. Dans notre cas, quand une paire d'objets est liée dans le classement établi (a obtenu la même note, ou le même score), cela signifie que l'utilisateur est en fait indifférent entre les deux. Ainsi, un système parfait ne doit pas classer un objet comme étant supérieur à l'autre.

Dans le cas où nous disposons seulement de données d'utilisation, il peut être approprié de construire un classement de référence où les éléments choisis (achetés) par l'utilisateur sont classés au-dessus des éléments non choisis. Cependant, ceci n'est valable que si nous savons que l'utilisateur connaissait ces articles non choisis, de sorte que nous

pouvons en déduire que l'utilisateur a en effet préféré les éléments choisis plutôt que les articles non choisis, et ce en toute connaissance de cause.

VII.1.C.i. NDPM

Dans certains systèmes il est nécessaire de définir un ordre total entre les éléments, et ce sans prendre en compte la présence ou non d'éléments liés. Dans ce genre de systèmes, l'évaluation ne doit pas pénaliser un système qui ordonne un élément par rapport à un autre alors que ceux-ci sont liés dans l'ordre de référence.

Dans ce cas, on utilise la méthode Normalized Distance-based Performance Measure (NDPM ; [YAO 95]), définie comme suit :

Avec DI_{uo_i} correspondant à l'index (place dans la liste) déterminé, et RI_{uo_i} correspondant à l'index de référence pour un objet o_i et un utilisateur u . Dans cette méthode on évalue toutes les paires possibles d'éléments constituant la liste. Cette évaluation des paires d'éléments consiste à déterminer si l'ordre défini par le système entre les éléments de la paire évaluée est le même dans l'ordre de référence.

Exemple :

Prenons une liste de référence (et donc un ordre de référence) : A, B, C. Et prenons une liste ordonnée générée par le système : A, C, B. La méthode va évaluer les paires suivantes : AB, AC et BC.

Afin de définir NPDM il nous faut définir quelques valeurs :

$$C^+ = \sum_{ij} \text{sgn}(DI_{uo_i} - DI_{uo_j}) \text{sgn}(RI_{uo_i} - RI_{uo_j}) \quad (45)$$

$$C^- = \sum_{ij} \text{sgn}(DI_{uo_i} - DI_{uo_j}) \text{sgn}(RI_{uo_i} - RI_{uo_j}) \quad (46)$$

$$C^u = \sum_{ij} sgn^2(RI_{uo_i} - RI_{uo_j}) \quad (47)$$

$$C^s = \sum_{ij} sgn^2(DI_{uo_i} - DI_{uo_j}) \quad (48)$$

$$C^{u0} = C^u - (C^+ + C^-) \quad (49)$$

Donc C^u correspond au nombre de paires d'éléments pour lesquels l'ordre de référence définit un ordre entre les éléments, c'est-à-dire le nombre de paires d'éléments non liés (et idem pour, C^s mais avec l'ordre défini par le système). C^+ et C^- correspondent respectivement au nombre de paires d'éléments concordant et discordant. C'est-à-dire au nombre de paires d'éléments qui sont dans un ordre identique dans l'ordre de référence et l'ordre défini par le système. C^{u0} correspond au nombre de paires où le système de référence ne lie pas d'éléments, mais où le système le fait.

Si on reprend notre exemple :

La paire AB est concordante, la paire AC est concordante et la paire BC est non concordante.

NDPM est donc défini comme suit :

$$NDPM = \frac{C^- + 0.5C^{u0}}{C^u} \quad (50)$$

Ainsi, la mesure NDPM donne un score parfait de 0 à un système qui permet de prédire correctement chaque relation de préférence par rapport à l'ordre de référence. Le pire score de 1 est attribué à un système qui contredit toutes les relations de préférence par rapport à l'ordre de référence. Ne pas prédire une relation de préférence est seulement pénalisé à moitié par rapport au fait de le contredire. Et également, on ne pénalise pas un

ordre défini entre des éléments, qui sont liés dans l'ordre de référence (on ne connaît pas la préférence de l'utilisateur entre ces deux éléments).

VII.1.C.ii. *Mesures de corrélation*

Dans le cas où l'on possède un ordre total entre les éléments, et que lorsque deux éléments sont liés (cela signifie que l'utilisateur est en fait indifférent à la différence entre les deux objets.) l'on veut que le système ne donne pas d'ordre supérieur à l'un des deux éléments par rapport à l'autre, des mesures de corrélations (coefficient de corrélation) telles que celles de Spearman Rho ou Kendall Tau [KEN 38, KEN 45] sont utilisées. On remarque que ces mesures tendent à être très similaires dans la pratique [FRE 07].

Le coefficient de Spearman Rho est calculé de la façon suivante (d_i étant la différence entre le rang de l'observation i et celui de sa valeur, et n est le nombre d'éléments) :

$$\rho = 1 - \frac{6 \sum_{i=1}^n d_i^2}{n^3 - n} \quad (51)$$

Cependant la méthode Kendall Tau possède également une variante, Kendall Tau-b, qui effectue des ajustements afin de mieux prendre en compte les éléments liés, c'est pourquoi nous allons définir uniquement les méthodes Kendall Tau et Kendall Tau-b. On remarque que dans le cas où des éléments sont liés, chaque item lié devrait se voir assigner un rang moyen dans la liste, c'est-à-dire que si deux éléments sont liés à la 2ème et 3ème place, alors on leur assigne un rang de 2.5 [KEN 45].

Le coefficient de corrélation de Kendall Tau est défini comme suit :

$$\tau = \frac{C^+ - C^-}{\frac{1}{2}n(n-1)} \quad (52)$$

Il permet ainsi de mesurer à quel point l'ordre défini par le système est proche de l'ordre de référence. La valeur de ce coefficient est comprise entre -1 et 1. 1 correspond à un

ordre défini par le système égal à l'ordre de référence. Tandis que -1 correspond à l'ordre inverse de l'ordre de référence. 0 correspond à la situation où l'ordre défini par le système n'est pas corrélé avec l'ordre de référence.

Afin de mieux prendre en compte les cas où les éléments sont liés entre eux, une variante dite Kendall Tau-b a été définie comme suit :

$$\tau_b = \frac{C^+ - C^-}{\frac{1}{2} \sqrt{(n(n-1) - U_x)(n(n-1) - U_y)}} \quad (53)$$

Avec

$$U_x = \sum u_x(u_x - 1) \quad (54)$$

$$U_y = \sum u_y(u_y - 1) \quad (55)$$

Les symboles u_x et u_y correspondent respectivement au nombre d'apparitions de chaque valeur X et Y dans la liste d'éléments.

Pour donner quelques exemples, prenons une liste X définie par (A, B, C, D, E). Et une liste Y contenant les mêmes éléments, mais dans un ordre différent.

Coefficient de Kendall Tau-b par rapport à 2 listes	A, B, C, D, E
A, B, C, D, E	1
E, D, C, B, A	-1
A, B, C, E, D	0.8
A, B, E, C, D	0.6
A, C, D, E, B	0.4
A, C, E, D, B	0.2

Tableau 7. *Exemple de valeur de coefficients de Kendall Tau-b*

Comment on peut le voir dans le Tableau 7, la valeur de 1 correspondant au classement parfait, et -1 au classement parfait inversé, et 0 correspondant à un classement ne se rapprochant pas de l'ordre parfait. On remarque que le fait d'avoir une valeur de 0.8 correspond à une situation presque parfaite (E a été déplacé d'un rang). De même, une valeur de 0.6 correspond à une situation très proche du parfait (E a été déplacé de 2 rangs). La valeur de 0.4 correspond à une situation qui commence à être moins correcte, mais qui tout de même offre un ordre de qualité acceptable (B a été déplacé de 3 rangs). La valeur de 0.2 correspond à une situation peu correcte, en effet l'ordre commence à ne plus être proche de la situation parfaite (B a été déplacé de 3 rangs et E de 1 rangs).

Ainsi on peut affirmer que des valeurs supérieures ou égales à 0.4 peuvent être considérées comme étant bonnes et celles supérieures à 0.6 comme étant très bonnes.

VII.1.D. Méthode de K-cross fold

Pour effectuer des tests, on utilise fréquemment une méthode appelée K-cross fold. Cette méthode est utilisée dans le domaine de l'apprentissage pour effectuer des tests afin d'évaluer les capacités d'un algorithme d'apprentissage, et ce en réutilisant plusieurs fois la même base de test.

Voici son principe :

1. En fonction de k, on divise la base de données de test en k parties,
2. On sélectionne la première k partie, et on la met de côté, cette k partie sera utilisée en tant que base de données de validation, alors que les autres parties vont être utilisées en tant que base d'apprentissage,
3. On utilise dans le système, comme données en entrée, la base d'apprentissage que l'on vient de définir,
4. On teste ensuite le système sur la base de validation. Et grâce aux données obtenues, on va pouvoir utiliser diverses métriques pour évaluer le système.
5. On répète les opérations 2 à 4, en changeant à chaque fois la k partie utilisée, afin d'utiliser toutes les k parties en tant que base de validation.

Afin d'évaluer un système, cette méthode est souvent employée, et à l'issue des k validations, on calcule généralement la moyenne des résultats obtenus par les métriques employées.

VII.2. Sélection des outils d'évaluation

Pour les expérimentations mises en place, il nous a fallu choisir quelles méthodes de mesures nous allions employer.

Le système que nous souhaitons tester n'utilise pas de système de prédiction de notes, donc les mesures définies au chapitre VII.1.A, ne peuvent être utilisées.

Le système que nous mettons en place a pour but de fournir une liste d'éléments à proposer à l'utilisateur qui soit dans un ordre qui corresponde à ses préférences. Il vient en complément d'un système de recherche qui lui a pour but de récupérer des éléments pertinents pour l'utilisateur. C'est pourquoi les méthodes définies au chapitre VI.A.B, ne sont pas utilisées.

De plus, le système mis en place a pour but de construire un ordre total entre les différents éléments qui sont proposés à l'utilisateur. On remarque ainsi que les métriques définies au chapitre VII.1.C correspondent tout à fait. Or dans le système mis en place, on ne définit pas d'ordre entre deux éléments liés, et dans notre système, il est fréquent d'avoir des éléments liés. C'est donc naturellement que lors des tests mis en place, on utilise comme mesure de performance, la mesure de coefficient de corrélation Kendall Tau-b. Cette mesure de performance reflétant à quel point les recommandations sont bonnes, c'est-à-dire à quel point l'ordre des éléments qui sont proposés à l'utilisateur sont dans le bon ordre.

VII.3. Conclusion

On constate que de nombreuses méthodes sont disponibles en fonction du système qu'il est nécessaire d'évaluer. Mais la majorité du temps, une mesure s'impose naturellement en fonction de ses caractéristiques. Nous mettons en œuvre plusieurs mesures au sein de nos expérimentations. Elles permettent d'évaluer les performances de notre système dans différents cadres, et en particulier, de comparer les performances de notre système avec celles d'autres systèmes de recommandation.

Chapitre VIII. Expérimentations

Dans le but de tester les capacités de notre système, et en particulier à quel point nos recommandations sont correctes, nous avons mis en place plusieurs tests. Ces tests sont en 3 parties, une première où l'on a essayé de reproduire le comportement d'un système de recommandation très utilisé. Ensuite, nous avons testé les capacités de notre système dans plusieurs cadres. Et finalement, nous avons comparé les performances de notre système par rapport à celles offertes par d'autres systèmes.

VIII.1. Comparaison avec Amazon

Un premier test a été conduit afin de pouvoir obtenir un classement similaire à ce qui est fait par le système de recommandation d'Amazon et ceci en employant une base de données issue d'Amazon. Amazon utilisant l'un des systèmes de recommandations les plus utilisés et performant du marché [LIND et al. 03].

VIII.1.A. Description

Afin d'évaluer les performances de notre système, nous avons tenté de retrouver un classement obtenu par l'algorithme de recommandation d'Amazon. Pour ce faire, nous avons effectué plusieurs requêtes sur le site d'Amazon, ces requêtes concernent des genres de livre qu'utilise Amazon, par exemple : « programming » (programmation), « historical » (historique), « mystery » (mystère), « horror » (horreur), « science-fiction », ou encore « cooking food and wine » (cuisine et vins). Pour chacun de ces genres, nous avons récupéré les 100 premiers livres proposés et avons conservé l'ordre dans lequel ils étaient proposés. Nous avons considéré que cette liste était la liste avec un ordre parfait qu'il fallait pouvoir obtenir avec notre système de recommandation.

De plus, pour chaque livre, nous avons récupéré toutes les informations disponibles sur celui-ci, mises à disposition par Amazon.

VIII.1.B. Méthodologie

Afin de pouvoir évaluer les performances de notre système, nous avons utilisé notre algorithme Pairwise (car aucune note n'est disponible), et la méthode de K-cross fold ($k = 6$), ainsi que la méthode de corrélation Kendall Tau-b.

Nous avons modifié de manière aléatoire l'ordre des livres présents dans la liste utilisée (liste des livres de science-fiction par exemple) ; Nous appelons cette liste la liste initiale, et avons utilisé ces données afin de construire le profil de l'utilisateur.

Pour cela, nous avons commencé par lister 10 livres sélectionnés de manière aléatoire dans la liste initiale (sélection issue de la base d'apprentissage obtenue par la méthode K-cross fold).

Voici un schéma représentant le fonctionnement général du test (Figure 11):

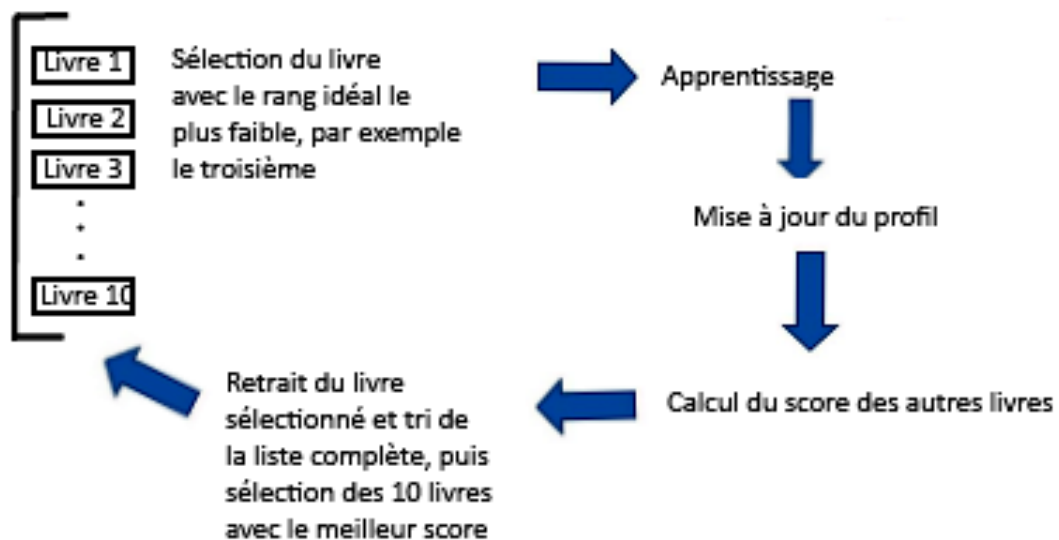


Figure 11. *Fonctionnement général du test*

1. Création d'une liste de 10 livres, à l'initialisation de manière aléatoire, puis par la suite, en sélectionnant les 10 livres ayant les meilleurs scores (voir étape 6),
2. À partir de cette liste de livres, nous avons considéré que l'utilisateur avait sélectionné le livre ayant le rang idéal (différent du rang actuel) le plus faible (le livre le plus intéressant pour l'utilisateur),

3. Nous avons lancé l'algorithme Pairwise en utilisant les livres issus de la liste créée à l'étape 1 et le livre sélectionné, conformément à la méthode Pairwise détaillée au chapitre V.5,
4. Nous avons ainsi mis à jour le profil de l'utilisateur, ensuite nous avons calculé un score pour chaque livre correspondant aux préférences de l'utilisateur
5. Nous avons trié la liste complète des livres de la liste (les livres de la liste initiale, moins ceux précédemment sélectionnés),
6. Nous avons sélectionné les 10 premiers livres de cette nouvelle liste,
7. Nous avons ensuite recommencé les opérations 1 à 6 jusqu'à avoir appris tous les livres présents dans la base d'apprentissage.

Puis nous avons calculé un score pour chacun des livres de la base de validation. À partir de ces scores, nous avons trié la base de validation.

Nous avons utilisé cette base triée afin de calculer la moyenne des écarts entre la place idéale de chaque livre et sa place déterminée par notre classement.

Nous avons également utilisé la méthode de corrélation de Kendall Tau-b afin de comparer la liste obtenue par notre système et cette même liste, mais ordonnée en fonction du rang initial de chaque livre.

VIII.1.C. Résultats

Nous avons utilisé la moyenne des écarts calculée. Qui est obtenue en utilisant la liste de livres de science-fiction (présentée ici, mais les autres listes ont donné des résultats similaires) et en faisant la moyenne des k tests avec une liste de 100 éléments.

Afin de confirmer les résultats de notre système, nous avons utilisé la moyenne de la valeur de Kendall Tau-b obtenue en utilisant notre algorithme Pairwise et la liste de livre de science-fiction pour les k tests.

Nous avons obtenu :

Moyenne des écarts	Coefficient de Kendall Tau-b
30.8	0.076

Tableau 8. *Moyenne des écarts et coefficient de Kendall Tau-b*

Ces valeurs obtenues (Tableau 8) sont très mauvaises, c'est pourquoi l'on a voulu en déterminer la cause, ainsi on a refait les calculs, mais cette fois en incluant dans la base d'apprentissage les données de la base de validation.

Nous avons ainsi obtenu :

Moyenne des écarts	Coefficient de Kendall Tau-b
26.9	0.116

Tableau 9. *Moyenne des écarts et coefficient de Kendall Tau-b en incluant la base de validation dans la base d'apprentissage*

En observant le Tableau 9, on constate que la situation est toujours mauvaise. En effet, le Coefficient de Kendall Tau-b de 0.116 correspond à une situation où les recommandations sont proches d'une situation aléatoire.

Nous avons alors pensé que la cause de ces mauvais résultats était la liste de livres utilisée qui serait trop homogène. Nous avons testé cela en utilisant une base contenant quelques livres de chacune des bases.

L'on a obtenu :

Moyenne des écarts	Coefficient de Kendall Tau-b
32.4	0.08

Tableau 10. *Moyenne des écarts et coefficient de Kendall Tau-b, avec des données issues de plusieurs bases d'apprentissage*

Cela est toujours très mauvais (Tableau 10), mais nous permet d'écarter un problème d'homogénéité des données. Ainsi, soit la base de données de test est mauvaise, dans le sens où il nous manque une information discriminante, soit l'algorithme est mauvais.

Afin de tester la base de données, nous avons utilisé le système Weka (cs.waikato.ac.nz/ml/weka) [HOL et al. 94], [FRAN et al. 10]. Ce système est une collection d'algorithmes d'apprentissage automatique pour des tâches de data mining (fouille de données). Weka contient des outils pour mettre en œuvre, en particulier, des méthodes de

classification (affecter des objets à des groupes) et de régression (peut être utilisé pour faire de la prédiction de notes). Il est également bien adapté pour le développement de nouveaux systèmes d'apprentissage.

Afin d'effectuer les tests, nous avons discrétisé la base de données et effectué divers tests en utilisant la base de données à travers plusieurs méthodes d'apprentissage (de régression et de classification).

Ces tests ont tous été mauvais, avec des valeurs de Kendall Tau-b allant de -0,07 à 0,11.

Nous avons ensuite pensé qu'il était possible que nous ayons mal discrétisé la base de données. Nous avons donc modifié notre base en utilisant le plus d'informations possible. Mais nous avons obtenu des résultats similaires. Donc ce n'est pas la discrétisation des données qui est la cause des mauvais résultats.

Après tous ces tests, nous avons pu déterminer que notre base initiale était la cause de ces mauvais résultats et qu'il nous manquait des informations discriminantes pour retrouver l'ordre présenté sur le site internet d'Amazon (*amazon.fr*). En effet même en utilisant d'autres techniques d'apprentissage et en exploitant la base de données, tous les résultats obtenus ont été mauvais.

VIII.1.D. Conclusion

Les mauvais résultats obtenus initialement nous ont guidés vers les idées que : soit le système proposé ici était mauvais, soit la base de données utilisée était mauvaise. La mauvaise qualité de la base de données a été démontrée à travers plusieurs tests, ainsi nous avons recherché une autre base de données.

C'est dans cette optique de recherche de nouvelle base de données que nous avons utilisé la base de données issue de « MovieLens ».

VIII.2. MovieLens (Base de données de test)

Nous utilisons un ensemble de données sur des films issus de la base MovieLens (accessible à l'adresse : « movielens.org »), qui est un service gratuit offert par le groupe de recherche GroupLens à l'Université du Minnesota. Cette base de données (BDD) est très utilisée dans le domaine de la recommandation [MIL et al. 03] et ne pose pas de problèmes contrairement à celle précédemment utilisée. Cette base de données contient des données tirées des actions (donner une note à un film) de vrais utilisateurs. Cependant, ces utilisateurs n'ont pas utilisé directement notre système.

Des tests sont réalisés sur cette base de données de films enrichie au niveau de la quantité d'informations qu'elle contient. Ces tests ont pour but d'évaluer les performances du système mis en œuvre dans cette thèse et particulièrement de comparer les deux algorithmes mis en œuvre, ainsi que la capacité du système à prendre en compte de manière efficace de nombreux attributs.

VIII.2.A. Comparaison entre les algorithmes 1 et 2

Nous testons les performances entre les méthodes « Rate Calculation » et « Pairwise Calculation ». Ces méthodes correspondent aux méthodes décrites respectivement aux chapitres V.4, et V.5.

La méthode « Rate Calculation » utilise des données explicites fournies par l'utilisateur, à savoir, une note. Et elle utilise cette note afin de faire évoluer le profil de l'utilisateur. Cette évolution se traduit par des modifications dans les scores attribués pour chaque attribut constituant le profil de l'utilisateur.

La méthode « Pairwise Calculation » utilise des données implicites, ces données correspondent à la liste des objets affichés à l'utilisateur et de l'objet que l'utilisateur a sélectionné. Grâce à ces données, une méthode de calcul basée sur du pair à pair, et sur des différences de score, permet de faire évoluer le profil.

VIII.2.A.i. Description

Nous voulions une base de données multicritère avec des notes de la part des utilisateurs et également que chaque utilisateur ait noté plusieurs objets.

C'est pourquoi nous avons utilisé la base de données issue de MovieLens. En effet, cette base de données correspond tout à fait à ce que nous recherchons et de plus, elle est souvent utilisée dans le but d'étudier certains systèmes de recommandation. Les membres de MovieLens l'utilisent, et ce, afin d'apprendre à construire de meilleurs systèmes de recommandation. Cette base de données contient 10 millions de notes et 100 000 tags (informations supplémentaires données par les utilisateurs sous forme de mots-clés) pour 10 681 films et pour 71 567 utilisateurs. Les évaluations sont basées sur une échelle de nombre entier allant de 1 à 5.

De plus, nous avons récupéré la base de données IMDB (Internet Movie DataBase, accessible à l'adresse : « *imdb.com* »). Cette base de données contient un très grand nombre d'informations à propos des films, comme les genres, auteurs, producteurs, réalisateurs, acteurs, réalisateurs effets spéciaux, *etc.* Cette base de données nous permet d'augmenter le nombre d'informations disponibles sur un film. Nous l'utilisons dans le but d'ajouter des informations dans notre système afin de pouvoir prendre en compte plus d'attributs de films.

VIII.2.A.ii. Méthodologie

Dans le but de tester notre système, un échantillon de 92 personnes a été sélectionné et nous avons effectué une série de calculs avec nos deux algorithmes. Ces utilisateurs ont été sélectionnés de manière à avoir des utilisateurs ayant fourni beaucoup de notes et également d'autres utilisateurs ayant donné peu de notes. Les notes données reflètent les évaluations qu'on fait les utilisateurs afin de définir leur degré de satisfaction par rapport aux films proposés. Dans ce test, nous prenons uniquement en compte les critères suivants : année de distribution, durée et les styles sans limiter leur nombre.

Pour effectuer ce test, une validation de type K-cross fold a été utilisée (avec $k = 6$) afin de pouvoir réutiliser notre base de données plusieurs fois. Pour effectuer la comparaison des performances entre nos algorithmes, nous utilisons pour chaque utilisateur la méthode de Kendall Tau-b. Afin de réaliser ce test, nous avons créé une base de données d'apprentissage et de validation grâce à la méthode de K-cross fold.

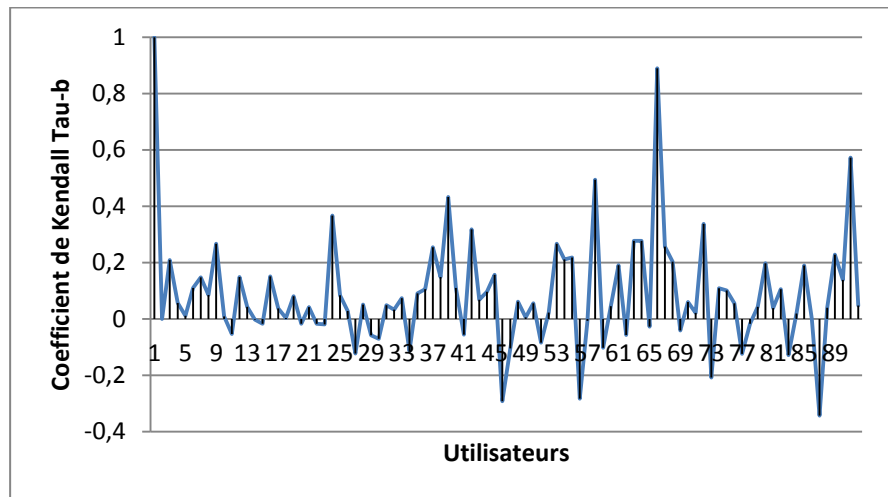
La méthode utilisée pour le test est la suivante (voir figure 8, mais utilisée avec des films à la place des livres) :

1. Création d'une liste de 10 films, à l'initialisation de manière aléatoire à partir de la base d'apprentissage, puis en sélectionnant les 10 films ayant les meilleurs scores (voir étape 6),
2. À partir de cette liste de films, nous avons considéré que l'utilisateur avait sélectionné le premier film ayant la note la plus élevée de la liste proposée (le film le plus intéressant pour l'utilisateur),
3. Nous avons lancé l'algorithme Pairwise en utilisant les films issus de la liste créée à l'étape 1 et le film sélectionné, conformément à la méthode Pairwise détaillée au chapitre V.5,
4. Nous avons ainsi mis à jour le profil de l'utilisateur, ensuite nous avons calculé un score pour chaque film correspondant aux préférences de l'utilisateur, et ce dans la liste initiale de films (moins ceux précédemment sélectionnés),
5. Nous avons trié la liste initiale (moins ceux précédemment sélectionnés),
6. Nous avons sélectionné les 10 premiers films de cette nouvelle liste,
7. Nous avons ensuite recommencé les opérations 1 à 6 jusqu'à avoir appris (sélectionné) tous les films présents dans la base d'apprentissage,
8. Puis nous avons calculé un score pour chacun des films de la base de validation,
9. À partir de ces scores, nous avons trié la base de validation et l'avons transformée en liste,
10. Nous avons remplacé chaque film de cette liste par leur note (exemple : 5 ; 4 ; 5 ; 3 ; 2),
11. Nous avons ensuite construit une seconde liste à partir de cette liste de validation, mais en triant les films par leur note de manière descendante, et en remplaçant de la même manière les films par leur note (exemple : 5 ; 5 ; 4 ; 3 ; 2),
12. Puis nous avons utilisé la méthode de Kendall Tau-b pour calculer le coefficient de corrélation entre ces deux listes

De cette manière, on peut déterminer la performance de notre système de recommandation, puisque plus ce coefficient est élevé et plus les recommandations produites sont considérées comme étant bonnes.

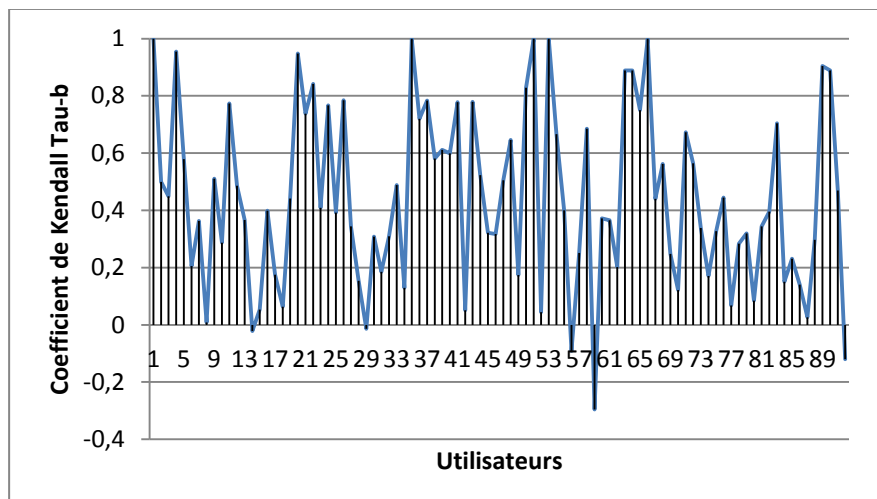
VIII.2.A.iii. Résultats

Premièrement, on a effectué des tests pour mesurer la performance de notre algorithme Rate :



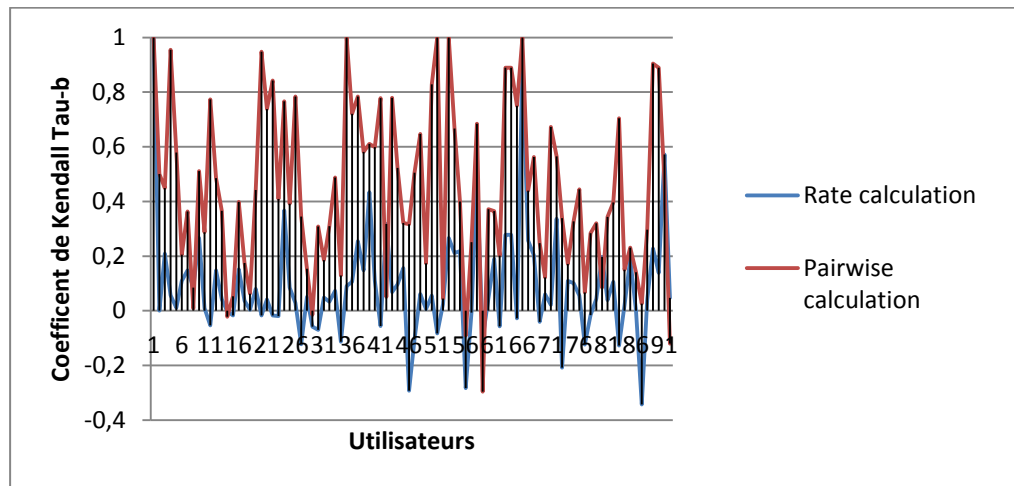
La moyenne de la valeur de Kendall Tau-b obtenue pour tous les utilisateurs est d'environ **0.09** (Graphique 1). Ce qui correspond à de mauvaises performances. Pour rappel on considère que les performances sont moyennes vers 0.2, bonnes vers 0.4, et excellentes à partir de 0.6.

Nous avons ensuite testé les performances de notre algorithme Pairwise :



La moyenne de la valeur de Kendall Tau-b obtenue pour tous les utilisateurs est d'environ **0.443** (Graphique 2). Ce qui correspond à de bonnes performances.

Afin de pouvoir mieux visualiser et comparer les performances de chaque algorithme, nous les avons regroupées :



Graphique 3. *Comparaison de la valeur de Kendall Tau-b obtenue par utilisateur par notre système mettant en œuvre l'algorithme Rate et Pairwise*

Nous pouvons déduire de ce graphique 3 que notre algorithme de comparaison par paires obtient une valeur de Kendall Tau-b supérieure à l'algorithme par note. Ce qui signifie que notre algorithme Pairwise offre de meilleures performances que celui par note (Rate).

VIII.2.A.iv. Conclusion

Nous pouvons déduire de ces tests que notre algorithme de comparaison par paires est bien meilleur que celui par note. En outre, nous pouvons dire que notre système fait plutôt de bonnes recommandations (la moyenne de Kendall Tau-b est d'environ 0.443, par rapport à un classement absolu parfait de Kendall Tau-b à 1 sur tous les objets présentés). Notre système de recommandation peut donc être considéré comme étant efficace.

VIII.2.B. Évaluation de la prise en compte de nombreux critères

Nous vérifions si notre système utilise de manière efficace de nombreuses données. Pour cela, nous lançons notre calcul par paire sur le même ensemble de données, mais nous changeons la quantité de critères pris en compte. Nous ne testons pas les performances de l'algorithme par note étant donné ses performances par rapport à l'algorithme Pairwise.

VIII.2.B.i. Données

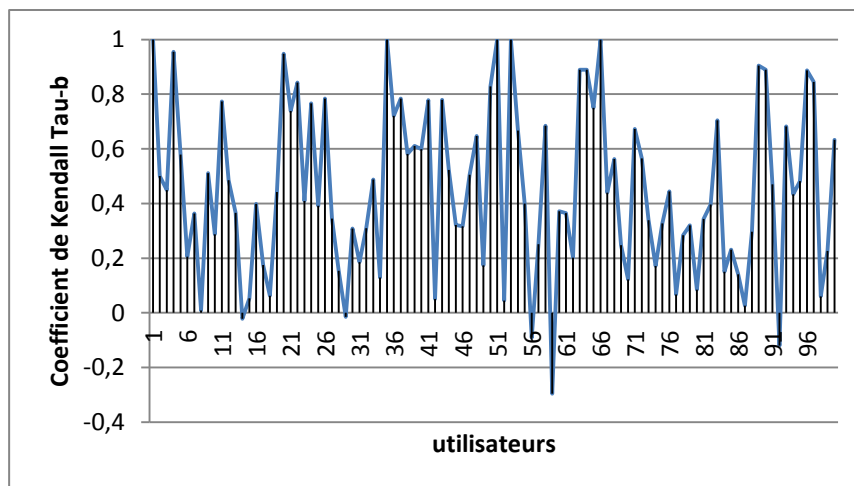
Les données sont les mêmes que celles utilisées précédemment.

VIII.2.B.ii. Méthodologie

Nous utilisons la même méthodologie que précédemment sauf que pour ce test nous avons utilisé 100 utilisateurs. De plus, d'une part, nous ne prenons en compte que l'année de la distribution de films, de nombreux styles (sans limite) et la durée. D'autre part, nous prenons en compte toutes les données disponibles (jusqu'à 21 critères différents).

VIII.2.B.iii. Résultats

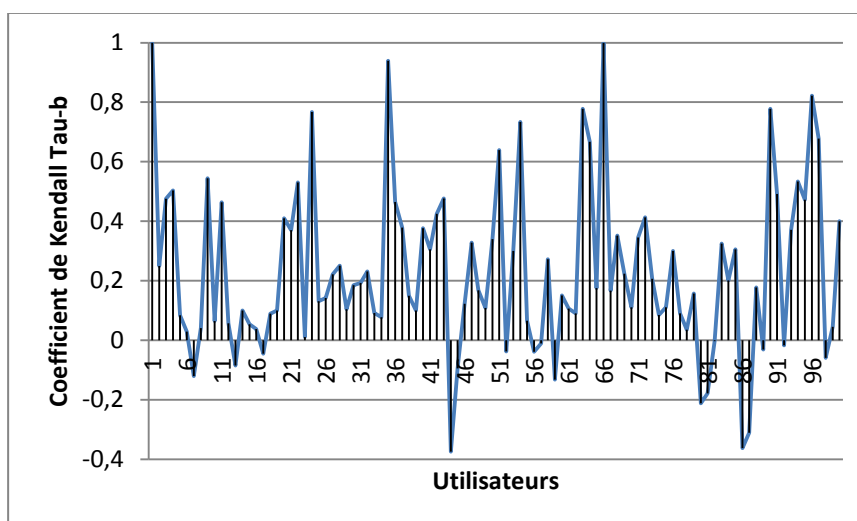
Premièrement, on a effectué des tests pour mesurer la performance de notre algorithme Pairwise avec une base de données restreinte :



Graphique 4. *Valeur de Kendall Tau-b obtenue par utilisateur par notre système mettant en œuvre l'algorithme Pairwise avec la base de données restreinte*

La moyenne de la valeur de Kendall Tau-b obtenue pour tous les utilisateurs est d'environ **0.45** (Graphique 4). Ce qui correspond à de bonnes performances.

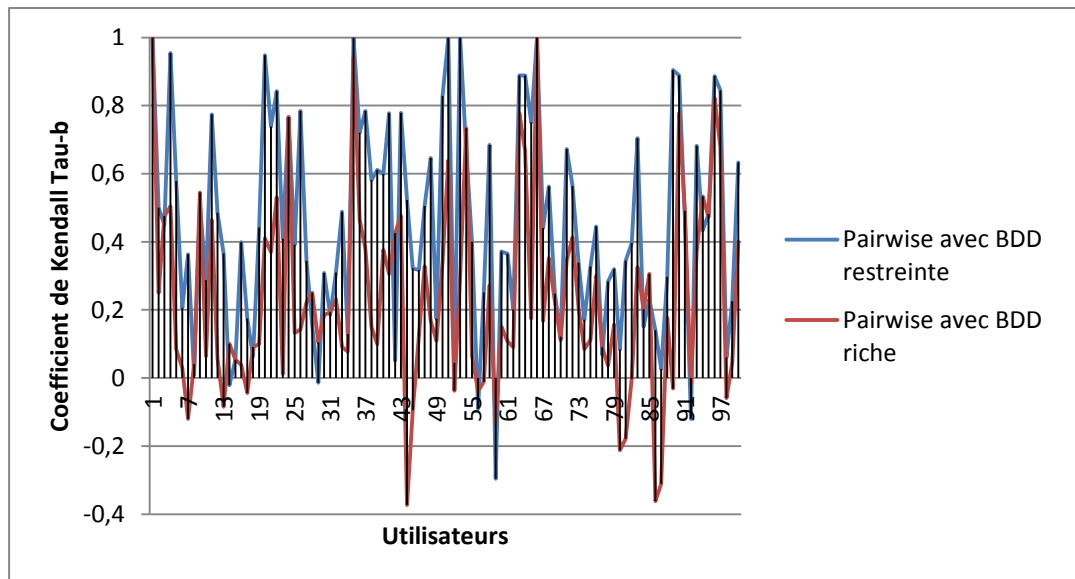
Puis, l'on a effectué des tests pour mesurer la performance de notre algorithme Pairwise avec une base de données riche :



Graphique 5. *Valeur de Kendall Tau-b obtenue par utilisateur par notre système mettant en œuvre l'algorithme Pairwise avec la base de données riche*

La moyenne de la valeur de Kendall Tau-b obtenue pour tous les utilisateurs est d'environ **0.233** (Graphique 5). Ce qui correspond à des performances moyennes.

Afin de pouvoir mieux visualiser et comparer les performances de chaque base de données (restreinte et riche), nous les avons regroupées :



Graphique 6. *Comparaison de la valeur de Kendall Tau-b obtenue par utilisateur par notre système mettant en œuvre l'algorithme Pairwise, pour une base de données restreinte et une base de données riche*

À partir de ce graphique 6, nous pouvons déduire qu'utiliser plus de critères n'a pas permis d'avoir de meilleurs résultats, au contraire. Ainsi, on a pu démontrer que le système tel qu'il est actuellement ne permet pas de bien prendre en compte de nombreuses informations.

On pourrait se demander s'il serait utile de mettre en œuvre une heuristique permettant d'utiliser, dans certaines conditions définies à l'avance, s'il vaut mieux utiliser une quantité réduite d'informations ou toutes les informations disponibles. C'est ainsi que nous avons comparé les performances de chacune des méthodes (avec restriction et sans restrictions) afin de déterminer si cela était utile.

Ainsi, on a eu un meilleur résultat avec la méthode utilisant tous les critères disponibles dans 22 % des cas, et cela a permis de passer d'une valeur de Kendall Tau-b de 0.45 à 0.465 (si on prend systématiquement la meilleure valeur entre les deux méthodes), soit un gain de seulement 1.5 % pour mettre en place une heuristique difficile à définir.

VIII.2.B.iv. Conclusion

Ainsi on a pu démontrer que l’algorithme de Pairwise est largement meilleur que l’algorithme de Rate. De plus, on a pu démontrer que notre système avait des difficultés lorsqu’on utilisait de nombreuses données, et également que mettre en place une heuristique déterminant dans quelles conditions on utilise tous les critères disponibles ou seulement une quantité réduite n’était pas la bonne solution. Une autre possibilité serait de changer notre système afin de l’améliorer lorsqu’il fonctionne avec plus de données, c’est ce qui est fait en utilisant les méthodes détaillées au chapitre VI. Cependant, pour des raisons techniques, nous n’avons pas pu mesurer le gain de ces méthodes lors de leurs utilisations au sein de notre système.

VIII.3. Mobifiz (Plateforme réelle)

Nous avons mis en place un test mesurant les performances de notre système par rapport à de vrais utilisateurs interagissant avec une plateforme en ligne (Mobifiz, *mobifiz.fr*) mettant en œuvre notre système. Cela est utile afin de vérifier les performances dans un cadre d’utilisation réelle afin de ne pas avoir uniquement une validation empirique sur, certes, des données issues de l’activité de vrais utilisateurs, mais sans utiliser directement notre système.

VIII.3.A. Description

Ainsi, nous avons déployé, grâce à des services web, notre système sur une plateforme en ligne actuellement encore en bêta (voir Annexes, Partie V). Cette plateforme propose des quiz jouables aux utilisateurs. À travers l’interaction des utilisateurs avec la plateforme, leurs notes et leurs sélections, nous pouvons construire le profil des utilisateurs. Notre système est ainsi utilisé lorsque les utilisateurs fournissent une note sur un quiz. Il est également utilisé lorsqu’un utilisateur sélectionne un quiz dans une liste de quiz qui lui est proposée. En effet, toutes les listes qui sont proposées à l’utilisateur sont triées en fonction de son profil. De cette manière, nous pouvons obtenir de nombreuses informations à propos des préférences de l’utilisateur en utilisant fréquemment l’algorithme Pairwise.

VIII.3.B. Méthodologie

Sur la page d'accueil, on trouve deux listes de 5 éléments (listes 1 et 2). Une de ces listes est aléatoire (liste 1), et l'autre sélectionne les quiz correspondant le mieux au profil de l'utilisateur (liste 2). Les utilisateurs ont également la possibilité de choisir un quiz dans une liste qui leur est présentée dans la partie solo du site, cette partie correspond à une sélection triée de quiz à faire en seul. Cela, après qu'ils aient choisi une catégorie majeure, et ensuite une sous-catégorie de quiz. L'ordre des quiz est pris en compte, ainsi que l'élément sélectionné par l'utilisateur.

Les utilisateurs sont divisés en 3 groupes. Les groupes 1, 2, 3 sont de taille équivalente. Un utilisateur ne peut faire partie que d'un seul groupe.

- Le groupe 1 correspond aux utilisateurs qui se voient proposer des recommandations basées uniquement sur l'apprentissage utilisant les notes qu'ils ont données aux quiz auxquels ils ont joué.
- Le groupe 2 correspond aux utilisateurs qui se voient proposer des recommandations basées uniquement sur l'apprentissage utilisant l'apprentissage de type Pairwise qui effectue l'apprentissage en fonction des quiz et de leur index lorsqu'ils sont présentés à l'utilisateur et surtout de l'élément sélectionné par l'utilisateur.
- Le groupe 3 correspond aux utilisateurs qui se voient proposer des recommandations basées sur l'apprentissage en fonction des notes données aux quiz, et en fonction de l'apprentissage Pairwise.

Tous les index des quiz choisis par l'utilisateur, dans la liste triée qui lui est proposée, sont enregistrés.

Ce test est effectué en prenant en compte l'activité des utilisateurs sur la plateforme et ce pendant 15 jours. Étant donné le statut bêta de la plateforme, et donc que la communication n'a pas encore été faite pour faire venir de nombreux utilisateurs sur la plateforme, la quantité d'utilisateurs est très réduite. Nous n'avons ainsi que 15 utilisateurs ayant en tout effectué 142 sélections sur la plateforme.

Les tests sont effectués en comparant les 3 groupes d'utilisateurs. Afin de déterminer l'impact de l'algorithme de désagrégation utilisé par le système de recommandation dans leur choix de quiz. Le principe étant que si le système recommande efficacement, alors l'utilisateur sélectionnera un quiz ayant un index faible dans la liste qui lui est présentée. En effet, la liste étant triée, le premier quiz est celui ayant obtenu le meilleur score en fonction de

ses caractéristiques et du profil de l'utilisateur. Les suivants sont ainsi considérés comme étant moins intéressants pour l'utilisateur, car ayant obtenu un score inférieur.

C'est donc grâce à l'index de l'élément sélectionné par l'utilisateur par rapport à la taille de la liste présentée, que la comparaison va pouvoir se faire.

Pour effectuer les comparaisons, nous avons utilisé la formule suivante :

$$IndicePerf_{eug}^{(t)} = 1 - \frac{index_{eug}^{(t)} - 1}{nb\ éléments\ affiché_{ug}^{(t)} - 1} \quad (56)$$

- e correspond à l'élément sélectionné
- u correspond à l'utilisateur actuel appartenant au groupe g
- g correspond au groupe de l'utilisateur
- t correspond au temps où l'élément a été sélectionné
- $index_{eug}^{(t)}$ varie entre 1 et $nb\ éléments\ affiché_{ug}^{(t)}$

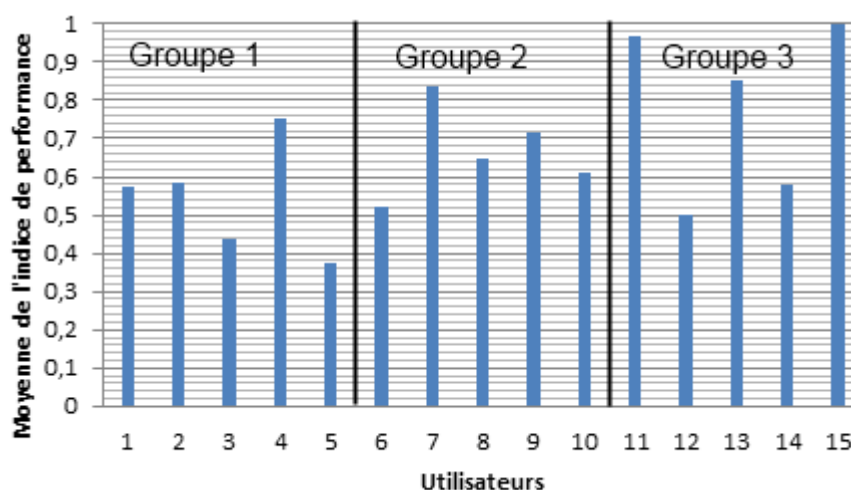
L'indice de performance $IndicePerf_{eug}^{(t)}$ est compris entre 0 et 1. 1 représentant la performance parfaite, c'est-à-dire, d'avoir positionné en première place le quiz sélectionné par l'utilisateur et 0 le fait d'avoir positionné l'élément sélectionné par l'utilisateur, en dernière position dans la liste. C'est un simple rapport entre l'index de l'élément sélectionné et le nombre d'éléments présenté à l'utilisateur, ce qui nous permet de mesurer l'écart entre l'index de l'élément sélectionné et la première place, ainsi que de prendre en compte le nombre de quiz présentés à l'utilisateur.

Afin de comparer les performances de chacun des 3 groupes, nous avons utilisé la moyenne des résultats obtenus par la méthode précédemment décrite.

Nous avons séparé la moyenne par utilisateur, et par groupe. Étant donné le nombre d'utilisateurs, nous n'avons pas effectué de calculs par rapport aux listes, car cela n'aurait pas été significatif.

VIII.3.C. Résultats

Les utilisateurs 1 à 5 font partie du groupe 1. Les utilisateurs 6 à 10 font partie du groupe 2. Les utilisateurs 11 à 15 font partie du groupe 3. Ainsi en effectuant la moyenne de performances pour chaque utilisateur, on obtient par utilisateur :



Graphique 7. *Moyenne de l'indice de performance par utilisateurs*

La moyenne de tous les utilisateurs est d'environ **0.663** (Graphique 7). Ce qui permet de dire que le système mis en place satisfait assez bien les utilisateurs. Par ailleurs, on voit ici que les résultats obtenus par le groupe 1 et 2 sont assez uniformes, alors que les résultats des utilisateurs appartenant au groupe 3 sont assez aléatoires. Ainsi, les méthodes Rate et Pairwise, obtiennent des résultats assez uniformes quand elles sont utilisées de manière séparée, à la différence de leur utilisation conjointe.

Ensuite nous avons déterminé la moyenne de l'indice de performance obtenu par groupe :

	Groupe 1 : Uniquement algorithme Rate	Groupe 2 : Uniquement algorithme Pairwise	Groupe 3 : Algorithme Rate ET Pairwise
Moyenne de l'indice de performance	0.544	0.665	0.778

Tableau 11. *Moyenne de l'indice de performance en fonction du groupe*

On peut ainsi constater sur le Tableau 11 que le l'algorithme Rate offre de moins bons résultats que l'algorithme Pairwise, apportant ainsi une confirmation des résultats obtenus dans MovieLens.

Par contre on obtient une information supplémentaire ici, à savoir qu'il est bénéfique d'utiliser les méthodes de manière conjointe. La satisfaction globale des utilisateurs ne sera, certes, pas uniforme, mais sera meilleure que l'utilisation d'uniquement l'une ou l'autre méthode.

VIII.3.D. Conclusion

Nous avons confirmé ici, par des tests avec des utilisateurs utilisant directement notre système, que notre système offrait un bon niveau de satisfaction aux utilisateurs. Nous avons également confirmé que notre algorithme Pairwise était meilleur que l'algorithme Rate pour fournir de bonnes recommandations. Par ailleurs, nous avons également pu démontrer que l'utilisation conjointe de ces deux algorithmes permettait d'obtenir des performances globales supérieures à celles obtenues avec les algorithmes utilisés séparément.

VIII.4. Comparaison avec un système multi-agents réalisant du profiling utilisateur

Afin d'évaluer notre système, nous nous sommes comparés, au niveau de la qualité des recommandations effectuées, avec un autre système permettant de faire de la recommandation grâce à un système de profiling utilisateur. Ainsi nous comparons les performances de notre système avec celles d'un système mettant en œuvre un système multi-agents afin de réaliser l'apprentissage de profils utilisateur. Les systèmes multi-agents sont exploités afin d'apporter une solution distribuée à la personnalisation des services offerts aux utilisateurs et à l'apprentissage de leurs centres d'intérêt. Cela est intéressant afin de pouvoir comparer les performances de notre système avec un système utilisant des méthodes de profiling différentes de celles habituellement employées (voir chapitre III).

VIII.4.A. Description

Dans le but de comparer les performances du système que nous présentons ici, nous avons mis en place un BQR (Bonus Qualité Recherche) avec d'autres membres du laboratoire mettant en œuvre des systèmes multi-agents dont l'un des buts, est de réaliser du profiling utilisateur. Le système multi-agents utilisé est issu des travaux de Sylvain Lemouzy [LEM 11].

VIII.4.B. Méthodologie

Les tests mis en œuvre consistent à évaluer les performances de l'algorithme Paiwise décrit ici, avec les performances du système multi-agents décrit dans [LEM 11].

Ces tests proposent d'évaluer la capacité des algorithmes à déterminer quels sont les films qui vont le plus satisfaire un utilisateur en fonction des caractéristiques des films en question (genre, auteur, producteur, réalisateur, acteurs, *etc.*).

Ces tests utilisent la base de données issue de MovieLens décrite au chapitre « VIII.2.A.i ». À partir de cette base, nous avons extrait deux bases d'apprentissage différentes :

- avec en entrée une base de données de films ne contenant qu'un nombre limité d'informations (BDD restreinte), en effet, les attributs pris en compte étaient : l'année de la distribution du film, de nombreux styles (sans limite), et la durée du film
- avec en entrée une base de données de films contenant un ensemble très complet de caractéristiques (BDD riche). Cette base incluant des informations comme le genre, auteur, producteur, réalisateur, acteurs, réalisateur effets spéciaux, *etc.*

Les tests s'appuient sur une sélection aléatoire de 100 utilisateurs issus de ces deux bases.

La méthode de comparaison utilisée pour mesurer les performances de chaque système est Kendall Tau-b. Cette méthode est utilisée dans le cadre de la méthode de K-cross fold (avec $k = 6$).

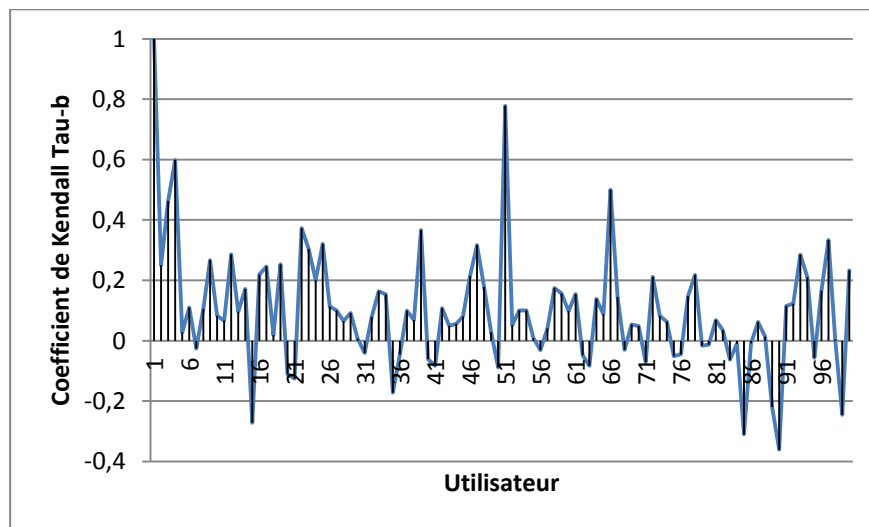
Chaque système a donc reçu en entrée une base, puis l'autre. Les systèmes ont utilisé ces données afin d'effectuer un tri (ranking) des films à présenter à l'utilisateur, et grâce à la méthode du K-cross fold, on a répété l'opération sur plusieurs sous-partie de la base initiale.

La moyenne des résultats obtenus pour chaque utilisateur et pour chaque K partie a été effectuée.

La procédure de test est identique à celle utilisée au chapitre VIII.2.A.ii.

VIII.4.C. Résultats

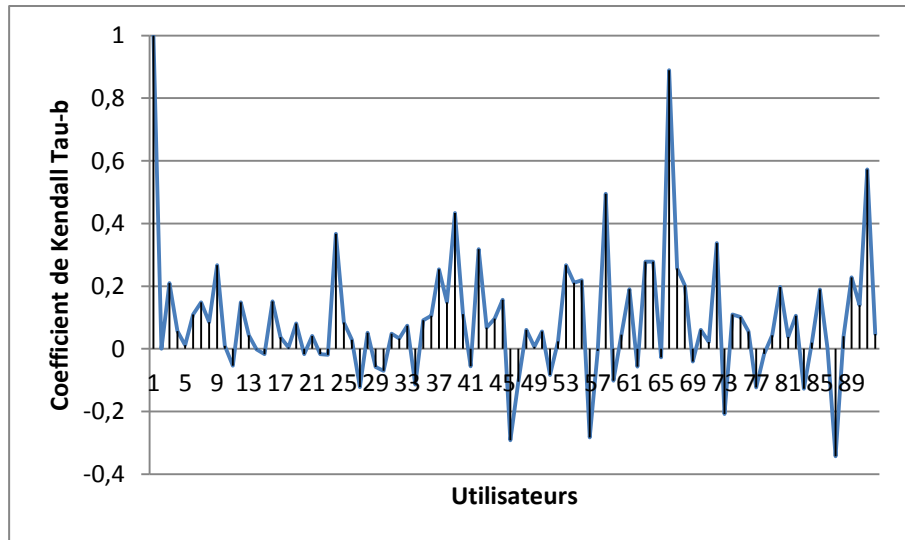
Premièrement les résultats issus du système multi-agents avec la base de données restreinte (BDD restreinte) :



Graphique 8. *Valeur de Kendall Tau-b obtenue par utilisateur par le système multi-agents utilisant une base de données restreinte*

La moyenne de la valeur de Kendall Tau-b obtenue pour tous les utilisateurs est d'environ **0.101** (Graphique 8). Ce qui correspond à de mauvaises performances. Pour rappel on considère que les performances sont moyennes vers 0.2, bonnes vers 0.4, et excellentes à partir de 0.6.

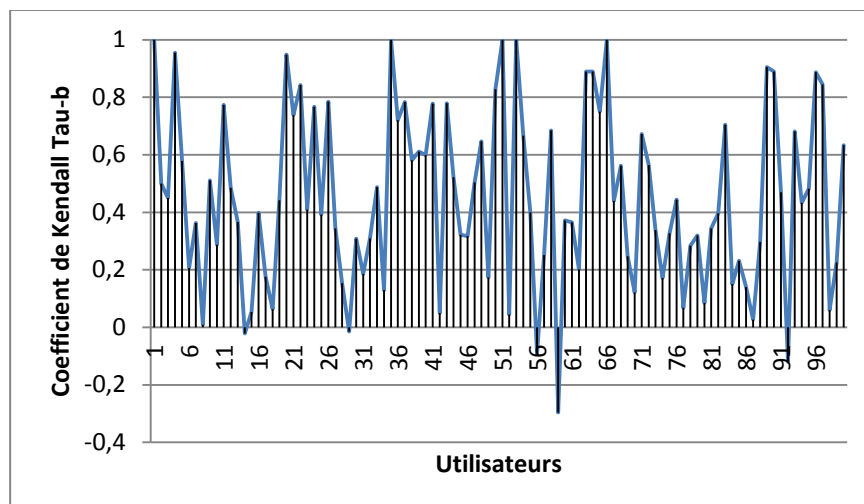
Voici les résultats issus du système présenté ici, et spécifiquement lors de la mise en œuvre de l'algorithme Rate en utilisant la base de données restreinte (BDD restreinte).



Graphique 9. *Valeur de Kendall Tau-b obtenue par utilisateur par notre système mettant en œuvre l'algorithme Rate*

La moyenne de la valeur de Kendall Tau-b obtenue pour tous les utilisateurs est d'environ **0.09** (Graphique 9). Ce qui correspond à de mauvaises performances.

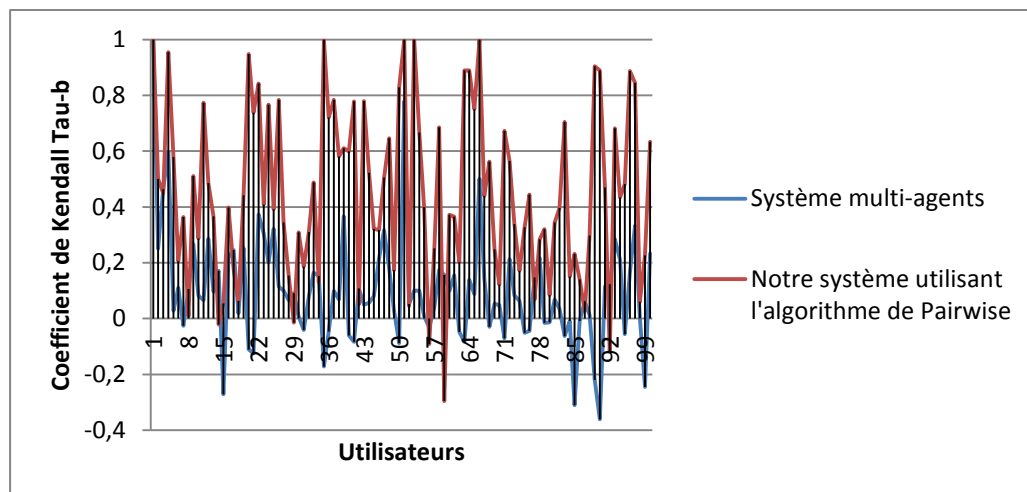
Voici les résultats issus du système présenté ici, et spécifiquement lors de la mise en œuvre de l'algorithme de Pairwise en utilisant la base de données restreinte (BDD restreinte).



Graphique 10. *Valeur de Kendall Tau-b obtenue par utilisateur par notre système mettant en œuvre l'algorithme Pairwise et utilisant une base de données restreinte*

La moyenne de la valeur de Kendall Tau-b obtenue pour tous les utilisateurs est d'environ **0.450** (Graphique 10). Ce qui correspond à de bonnes performances.

Afin de pouvoir mieux visualiser et comparer les performances de chaque système, nous les avons regroupées :

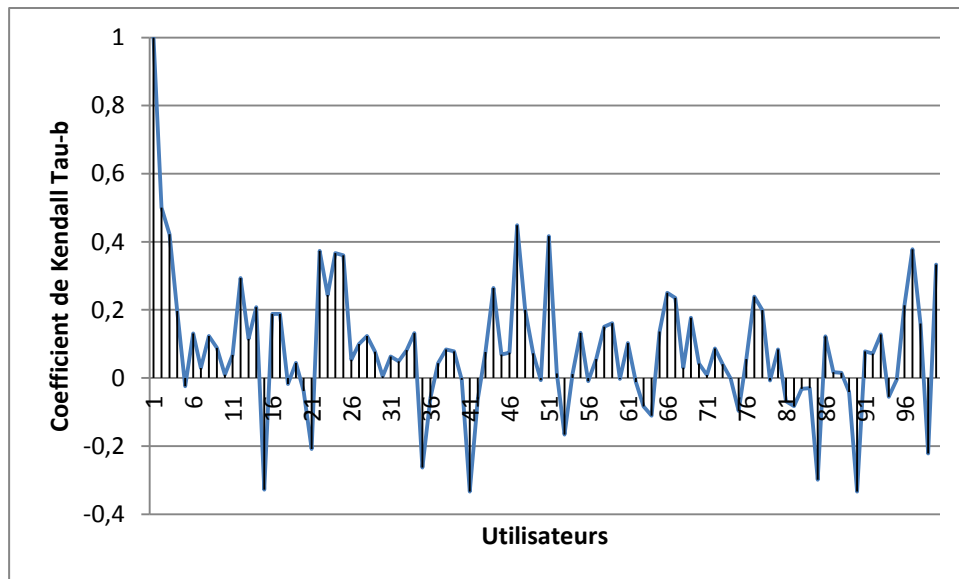


Graphique 11. *Comparaison des valeurs de Kendall Tau-b obtenue par utilisateur pour le système multi-agents et notre système utilisant l'algorithme de Pairwise avec une base de données restreinte*

On constate donc sur le graphique 11, que le système Pairwise développé dans cette thèse obtient une valeur de Kendall Tau-b supérieure à celle obtenue en utilisant un système multi-agents. On peut en déduire que le système de recommandation décrit ici offre de meilleures recommandations que le système multi-agents. On remarque cependant que lorsque l'on utilise l'algorithme Rate, le système multi-agents obtient de meilleures performances.

Après avoir effectué ces tests, nous avons comparé les performances obtenues par chacun des deux systèmes dans le cadre où de nombreuses données sont utilisées. C'est pourquoi nous avons fait de nouveaux tests en utilisant cette fois une base de données comportant de plus nombreuses informations détaillant les films.

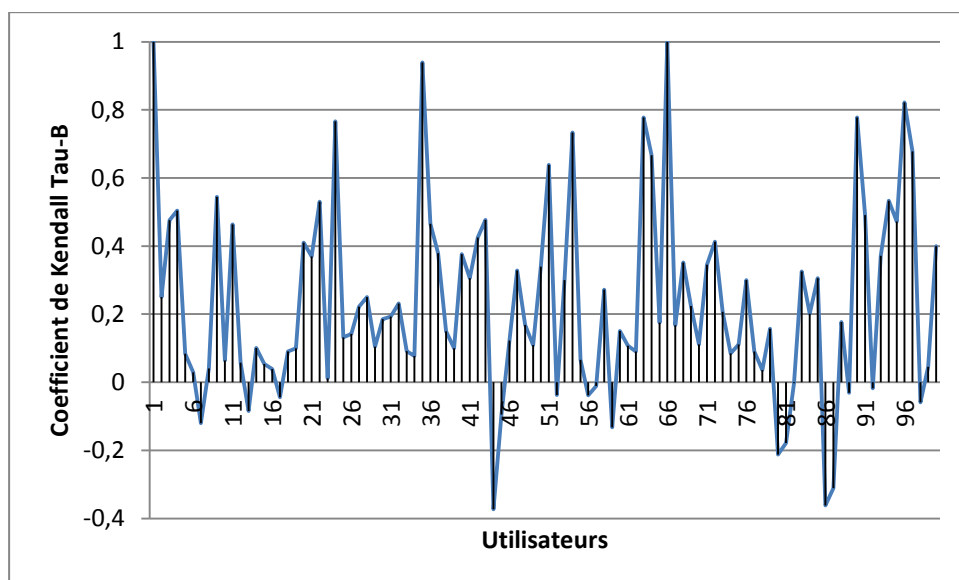
Voici les résultats du système multi-agents utilisant la base de données complète (BDD riche) :



Graphique 12. *Valeur de Kendall Tau-b obtenue par utilisateur par le système multi-agents utilisant une base de données riche*

La moyenne de la valeur de Kendall Tau-b obtenue pour tous les utilisateurs est d'environ **0.081** (Graphique 12). Ce qui correspond à de mauvaises performances.

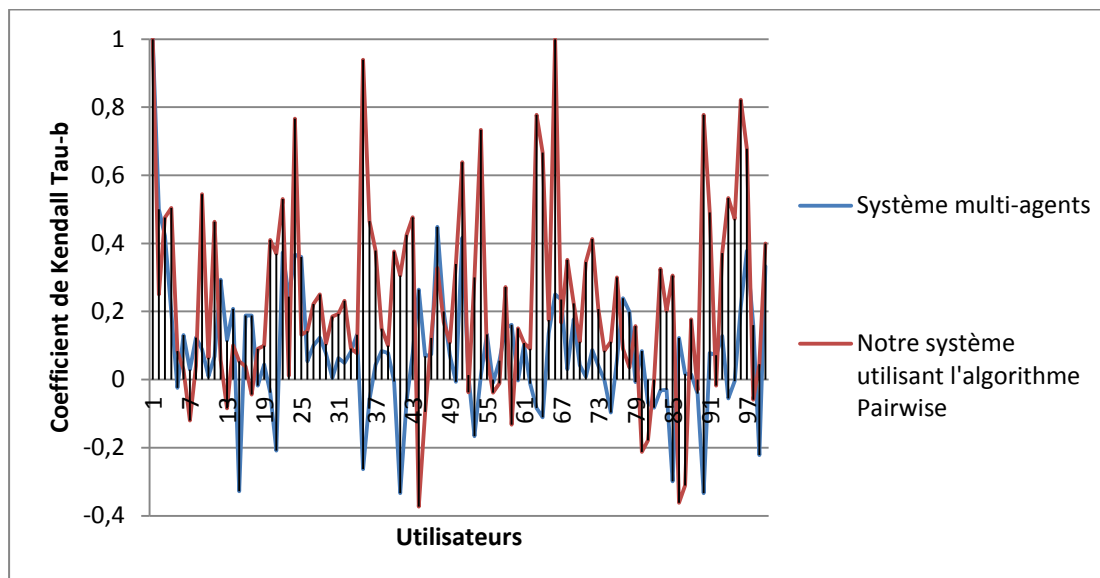
On utilise pour comparer ces résultats, les données issues de notre système utilisant lui aussi la base de données riche :



Graphique 13. *Valeur de Kendall Tau-b obtenue par utilisateur par notre système mettant en œuvre l'algorithme Pairwise et utilisant une base de données riche*

La moyenne de la valeur de Kendall Tau-b obtenue pour tous les utilisateurs est d'environ **0.233** (Graphique 13). Ce qui correspond à des performances moyennes.

Afin de pouvoir mieux visualiser et comparer les performances de chaque système, nous les avons regroupées :



Graphique 14. *Comparaison des valeurs de Kendall Tau-b obtenue par utilisateur pour le système multi-agents et notre système utilisant l'algorithme de Pairwise avec une base de données riche*

On lit sur ce graphique 14 que, comme précédemment, le système décrit ici et mettant en œuvre l'algorithme Pairwise obtient de meilleures performances que le système multi-agents.

Cependant on constate que les performances de notre système ont quasiment été divisées par 2 (0.45 à 0.233) lorsque l'on a pris en compte tous les critères disponibles. Et si on regarde les performances du système multi-agents, ses performances ont peu changé (0.101 à 0.081). Le système multi-agents semble donc mieux résister à l'augmentation du nombre de critères utilisés.

VIII.4.D. Conclusion

Après avoir observé ces différents résultats, on peut en conclure que notre système fournit de meilleures recommandations que celles obtenues par le système multi-agents. En effet, que cela soit avec une quantité réduite d'attributs pris en compte, ou en utilisant toutes

les informations disponibles à propos des films, on a montré que notre système était meilleur que le système multi-agents.

Cependant, le système employant un système multi-agents n'a pas été conçu dans l'objectif d'être un système de recommandation, mais dans le but d'effectuer de la personnalisation de l'accès à l'information. Cela peut expliquer les mauvais résultats dans le contexte présenté ici. Il faudrait poursuivre les expérimentations pour déterminer les performances et caractéristiques des systèmes multi-agents dans d'autres contextes.

Cela nous permet de fournir une première comparaison des performances de notre système avec d'autres systèmes adaptatifs.

VIII.5. Comparaison avec un système employant la recommandation collaborative

Dans le but de comparer les performances du système que nous proposons avec celles de systèmes mettant en œuvre d'autres techniques, nous nous comparons ici à un système mettant en œuvre de la recommandation collaborative. Ceci est important, car ces systèmes sont actuellement très employés, et cela nous permet, en fonction des performances de notre système, de déterminer sa place parmi les autres systèmes de recommandations.

VIII.5.A. Description

Afin de pouvoir comparer les performances de notre système, et en particulier l'algorithme de Pairwise que nous utilisons, nous nous comparons aux résultats obtenus dans l'article de Seung-won Hwang et Mu-Woong Lee [HWA 09] qui mettent en œuvre des algorithmes de recommandation collaborative (RC ou CF comme Collaborative Filtering). Dans cet article sont comparées les capacités du système que les auteurs ont mis au point afin d'utiliser des techniques de recommandation collaborative sans utiliser de notes fournies par les utilisateurs. En effet, le but de l'article est de comparer les performances d'un système à base de CF quantitatif (basé sur des notes) à des systèmes de CF qualitatifs (basé sur des feedback des utilisateurs). La technique de CF quantitatif employée se base sur le moteur open-source de recommandation collaborative CoFE (ce moteur n'est plus maintenu actuellement). Ce système a en particulier été utilisé par l'équipe du Yahoo! Research Labs travaillant sur les systèmes de recommandation [DEC et al. 05]. Dans ce système, la

technique utilisée pour réaliser le calcul de similitude entre les utilisateurs est celle de Pearson [LIN 89], cette mesure est proche des mesures de Kendall Tau et de Spearman Rho.

Or cela est particulièrement intéressant, car notre système utilise également une recommandation qui peut être considérée comme qualitative (le système Pairwise) dans lequel on utilise des feedback utilisateur (leurs sélections). Ainsi nous allons comparer les performances de notre système avec à la fois un système collaboratif basé comme nous sur les feedback utilisateurs et également avec un système mettant en œuvre de la recommandation collaborative classique.

VIII.5.B. Méthodologie

Pour le test de notre système, un échantillon de 100 personnes a été pris et nous avons lancé le calcul en utilisant l'algorithme Pairwise. Nous prenons en compte uniquement les critères suivants : année de distribution, durée et les styles sans limiter leur nombre. Pour effectuer ce test, la méthode K-cross fold (avec $k = 6$) a été utilisée. Pour effectuer la comparaison entre les systèmes, nous utilisons la moyenne des valeurs obtenues, calculées avec Kendall Tau (et non Tau-b, les auteurs ayant préféré la méthode Kendall Tau) pour chaque partie.

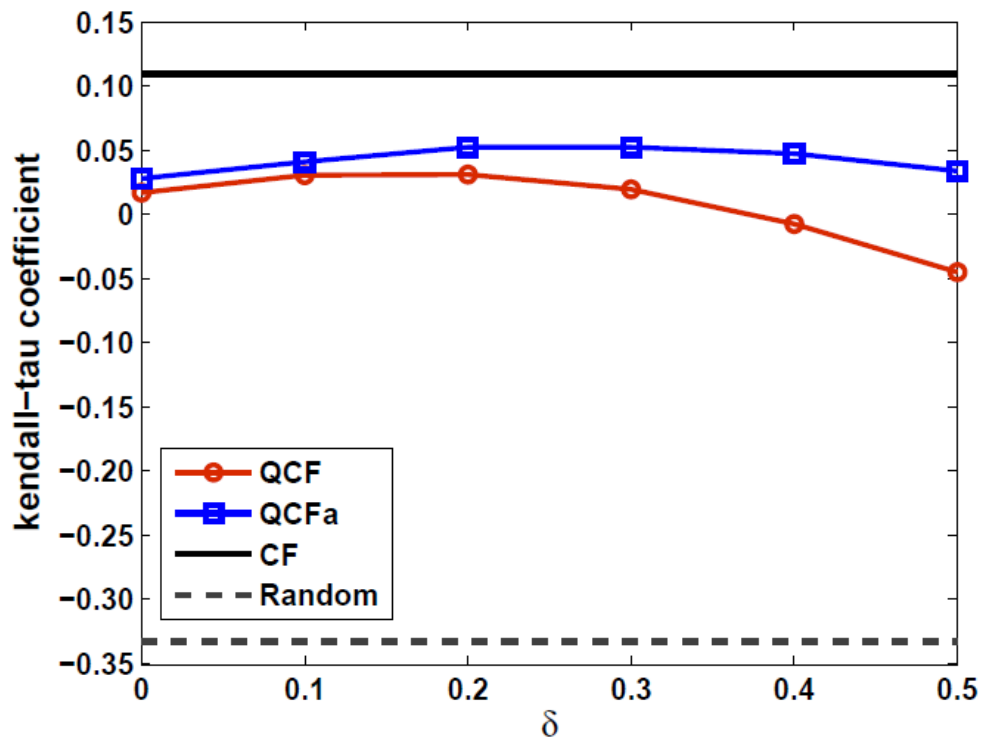
Pour le test du système détaillé dans l'article de Seung-won Hwang et Mu-Woong Lee [HWA 09], les auteurs mettent en œuvre des algorithmes de recommandation collaborative. Trois systèmes sont mis en œuvre, les résultats issus de ces systèmes ont été rassemblés dans le Graphique 15 et c'est avec celui-ci graphique que nous nous comparons.

La métrique utilisée pour leurs tests est celle de Kendall tau. Les tests se basent sur la base de données de MovieLens. La base utilisée est une version réduite de la base que nous utilisons nous même. Cette base est composée de 100.000 votes définis par 943 utilisateurs sur 1682 objets. Ensuite, ils ont sélectionné les 100 objets les plus notés. Pour chaque recommandation, ils ont regroupé les notes de 100 utilisateurs ayant des profils voisins. À partir de cette base, ils ont mis de côté 5% de notes sélectionnées au hasard, qui vont servir de base de validation pour les tests. L'expérimentation consiste à comparer la précision de leurs systèmes avec un système employant de la recommandation collaborative classique (quantitative, se basant sur les notes des utilisateurs). Cette recommandation met en œuvre le moteur CoFE. Pour la comparaison, ils calculent la valeur de Kendall tau pour chaque utilisateur et la moyenne des résultats obtenus. Ces résultats sont également comparés aux résultats obtenus avec des prédictions aléatoires.

VIII.5.C. Résultats

Ces résultats illustrent la précision des systèmes présentés par les auteurs (QCF et QCFA, qui représentent Quantitative Collaborative Filtering), du système de recommandations collaboratives plus classique (CF), ainsi qu'aux résultats de recommandations aléatoires.

Les résultats sont donnés en fonction de la valeur moyenne des résultats obtenus par la méthode de comparaison de Kendall tau, et également en fonction de δ , cette variable représente la différence entre deux valeurs. Cette différence est utilisée pour déterminer si des objets d'une paire d'objets donnée sont liés ou non. Pour rappel, quand une paire d'objets est liée, cela signifie que l'utilisateur est en fait indifférent à la différence entre les deux.



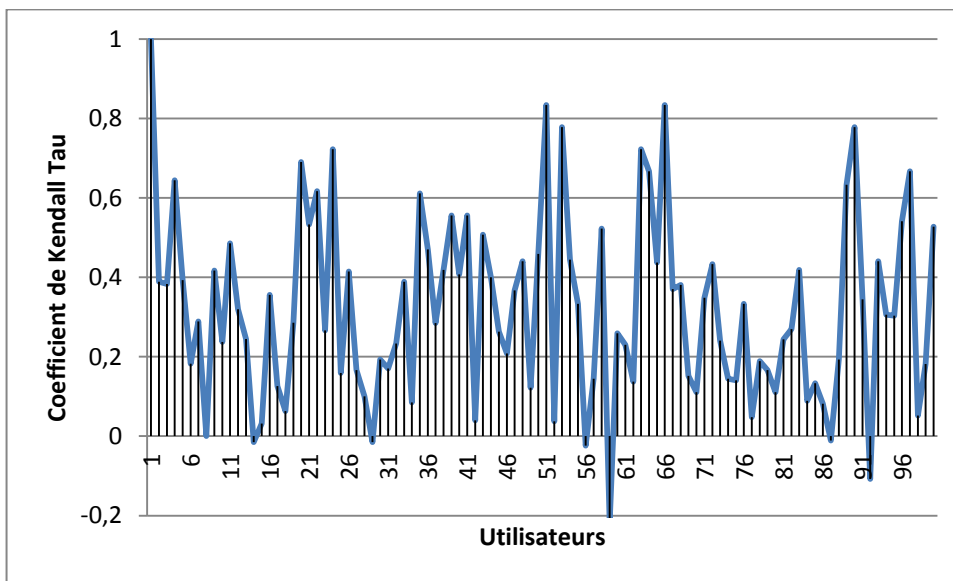
Graphique 15. *Valeur du coefficient de corrélation Kendall Tau QCF, QCFA, et CF en fonction de la valeur de δ*

On remarque sur ce graphique 15, que la moyenne de QCF est d'environ 0.05, celle de QCFA, environ 0.02, et la moyenne de la méthode CF est de 0.11. Ce qui correspond à de

mauvaises performances. Pour rappel on considère que les performances sont moyennes vers 0.2, bonnes vers 0.4, et excellentes à partir de 0.6.

Afin de comparer ces performances avec celles de notre système, nous avons calculé avec la méthode de Kendall tau et non Kendall tau-b (en effet, les auteurs utilisent uniquement Kendall tau), les performances de notre système lorsque l'on utilise l'algorithme Pairwise.

Dans le système que nous mettons en œuvre, la valeur de δ est toujours égale à 0.



Graphique 16. *Valeur de Kendall Tau obtenue par utilisateurs par notre système mettant en œuvre l'algorithme Pairwise et utilisant une base de données riche*

La moyenne de la valeur de Kendall Tau obtenue pour tous les utilisateurs est d'environ **0.32** (Graphique 16). Ce qui correspond à de moyennes/bonnes performances. Cela étant dû à l'absence de prise en compte des objets liés.

VIII.5.D. Conclusion

Bien que les techniques de recommandations collaboratives soient difficilement comparables avec des techniques de recommandation basées sur le contenu, on peut constater que les résultats obtenus par notre système sont meilleurs que les systèmes de recommandation collaborative qualitative présentés par les auteurs. Mais également, que

notre système se révèle meilleur que le système de recommandation collaborative plus classique utilisé et mis en œuvre dans CoFE. Cela permet de confirmer ses bonnes performances par rapport aux autres systèmes existants.

VIII.6. Comparaison avec un système de recommandation hybride

Afin de comparer les performances de notre système avec celles d'autres systèmes, nous nous comparons avec un système hybride [GAR 06]. Ce système utilise un mélange entre la recommandation collaborative et la recommandation sur le contenu. Une de ses caractéristiques est également de construire et d'exploiter une sémantique des caractéristiques.

VIII.6.A. Description

L'approche utilise des mesures de similarité entre les utilisateurs, mais elle propose aussi d'évaluer directement les attributs des éléments. Cela peut être utilisé afin de faire des recommandations directement aux utilisateurs, mais une autre caractéristique tout aussi importante, est que cela permet à ces recommandations d'être justifiées. Les auteurs introduisent une méthode pour prédire la préférence d'un utilisateur pour un film, en estimant l'attitude de l'utilisateur envers des caractéristiques avec lesquelles les autres utilisateurs ont décrit ce film. L'approche utilisée est hybride, au sein de laquelle un mécanisme de recommandation spécifique à l'utilisateur est appris et évalué expérimentalement. Il semble qu'un tel système de recommandation a la capacité de réaliser des améliorations significatives en ce qui concerne la précision de méthodes alternatives, et cela, tout en conservant d'autres avantages.

VIII.6.B. Méthodologie

L'évaluation des travaux des auteurs de [GAR 06], se fait vis-à-vis de l'évaluation de méthodes de recommandation collaborative (CF) issue de [HER et al. 04], de méthode hybride (Combined) décrite dans [GAR 06], d'une autre version du système hybride (Recommandz) décrite dans [GAR 05], une méthode dite POP (voir tableau 11) qui prédit

que chaque utilisateur va évaluer chaque élément avec la note globale moyenne de tous les autres utilisateurs, issu de [BREE et al. 98], et enfin de l'aléatoire (Random).

L'Ensemble des données du système utilisé dans [GAR 06] a été restreint aux utilisateurs avec 30 ou plusieurs notes, aux articles notés par au moins 5 utilisateurs. Cela a laissé 885 utilisateurs, 1641 articles, 667 caractéristiques, et un total de 19 621 notes.

La méthode d'évaluation utilise une méthode de corrélation, le coefficient de Spearman Rho. Ce coefficient tend à être très similaire à celui de Kendall Tau-b dans la pratique [FRE 07], c'est pourquoi nous pouvons le comparer directement avec les valeurs du coefficient de Kendall Tau-b obtenues par notre système

VIII.6.C. Résultats

Méthode	Coefficient de Spearman Rho
Combined	0.3523
CF	0.3214
Recommenz	0.3071
POP	0.3078
Random	0.0104

Tableau 12. *Comparaison avec des systèmes de recommandation hybride, en utilisant le coefficient de Spearman Rho*

Pour rappel, nous avons obtenu :

Dans le cadre de l'utilisation de l'algorithme de Kendall Tau-b sur une base de données restreinte, une valeur du coefficient de Kendall Tau-b de 0.450 (Graphique 10). Ce qui correspond à de bonnes performances.

Dans le cadre de l'utilisation de l'algorithme de Kendall Tau-b sur une base de données riche, une valeur du coefficient de Kendall Tau-b de 0.233 (Graphique 13). Ce qui correspond à des performances moyennes.

Ainsi, l'on constate sur le tableau 12, que dans le cas où nous limitons le nombre de critères, nous sommes meilleures que toutes les méthodes utilisées dans ce test. Par contre lorsque nous utilisons de nombreux critères, notre système est moins performant que ces autres méthodes.

VIII.6.D. Conclusion

Ainsi, ce test nous permet d'augmenter la quantité de tests permettant de comparer notre système avec ceux existants. Les résultats indiquent que notre système est efficace quand le nombre de critères est limité, et confirme que notre système est peu performant lorsque l'on utilise de nombreux critères. Cela était une information déjà connue pour laquelle la résolution est prévue dans les perspectives.

Conclusion générale des expérimentations

Après avoir effectué un descriptif des différents outils de mesure disponible pour la méthodologie, nous avons choisi les outils qui correspondent à notre cadre d'utilisation.

Les différentes expérimentations réalisées ont permis de comparer les performances de nos deux algorithmes de désagréations, et de pouvoir tester les performances de notre système sur une base de données contenant des informations à propos d'actions réalisées par de vrais utilisateurs (mise de notes), mais n'ayant pas utilisé directement notre système. Elles ont aussi permis d'évaluer les performances de notre système lors d'une utilisation directe par de vrais utilisateurs. Et enfin de comparer les performances que nous avons obtenues par rapport à d'autres systèmes, tels qu'un système employant un système multi-agents, un système utilisant des recommandations collaboratives quantitatives, qualitatives, ainsi que des systèmes utilisant une recommandation hybride.

Le test entre les performances de nos deux algorithmes a permis de montrer que notre algorithme Pairwise était meilleur que l'algorithme Rate.

Bien que l'échantillon soit limité, les tests réalisés avec de vrais utilisateurs utilisant directement notre système ont montré que notre système offrait de plutôt bonnes performances.

Les tests réalisés avec de vrais utilisateurs n'utilisant pas directement notre système (utilisateurs de MovieLens), ou d'autres systèmes de recommandation (multi-agents, collaboratif, hybride), nous permettent de valider le fait que notre système soit efficace.

Cependant, les tests réalisés afin de montrer si notre système utilisait de manière positive un grand nombre de critères ont révélé que notre système n'obtenait pas de bonnes performances dans ces cas-là. C'est pourquoi nous avons défini des méthodes utilisant des intégrales de Choquet afin de prendre en compte les interactions entre les critères dans le but d'améliorer les performances de notre système lors d'une utilisation mettant en œuvre une grande quantité de critères. Mais pour des raisons techniques, nous n'avons pas pu évaluer les résultats obtenus avec ces méthodes.

Les résultats des tests ont été rassemblés au sein d'un tableau récapitulatif (Tableau 13) :

	Caractéristiques de la Base de données utilisée	Résultats
Amazon	Utilisateurs simulés	Performances très mauvaises, problèmes avec la base de données
Movielens	Utilisateurs réels , mais sans utilisation directe de notre système	Bonnes performances avec une quantité réduite de critères, moins bons résultats avec de nombreux critères, mauvaise résistance au bruit
Mobifiz	Utilisateurs réels avec une utilisation directe de notre système	Bonnes performances en utilisation réelle. Intérêt démontré d'utiliser conjointement nos deux algorithmes de désagrégation
Recommandation multi-agents	Utilisateurs réels , mais sans utilisation directe de notre système	Performances supérieures au système multi-agents. Cependant, le système multi-agent a une meilleure résistance au bruit
Recommandation collaborative	Utilisateurs réels , mais sans utilisation directe de notre système	Performances supérieures aux systèmes collaboratifs testés
Recommandation hybride	Utilisateurs réels , mais sans utilisation directe de notre système	Performances supérieures au système hybride quand on utilise une quantité réduite de critères, sinon le système hybride est meilleur et les autres techniques également

Tableau 13. *Récapitulatif des tests effectués*

Il pourrait être intéressant de comparer les performances de notre système avec celles d'autres systèmes de recommandation utilisés plus largement (systèmes employés sur des sites E-Commerce très fréquentés) que ceux avec lesquels nous nous sommes comparés. C'est une de nos perspectives.

Conclusion générale de la thèse

Le système que nous avons mis au point satisfait les contraintes qui ont été définies dans la problématique de cette thèse.

- Ainsi nous avons mis au point un système qui est évolutif, c'est-à-dire que le ou les profils des utilisateurs évoluera/ont dans le temps, et l'assistance fournie évoluera de la même manière.
- Il met en œuvre les principes issus des domaines multicritère/multiattribut et cela en faisant varier dynamiquement les critères/attributs pris en compte pour la décision.
- Le système proposé est totalement incrémental, c'est-à-dire qu'il ne dispose pas de données *a priori* sur les utilisateurs, et de plus, il est synchrone.
- Il propose des recommandations génériques, c'est-à-dire qu'il peut être utilisé avec différents types d'éléments (une voiture, une maison, un film) et cela de manière naturelle.

Nous avons mis au point des techniques permettant d'apporter en partie des solutions à des verrous technologiques comme la désagrégation de critères et la prise en compte d'un nombre variable de critères dans le processus d'aide à la décision interactif, et ce sans définir au préalable de famille cohérente de critères sur laquelle est basée la décision.

Le système mis en place permet de mettre en œuvre un système supportant l'utilisation dynamique d'un nombre variable de critères, dans le but d'effectuer de l'assistance aux décisions de l'utilisateur.

Il permet d'utiliser des profils évolutifs qui offrent aux utilisateurs la possibilité de changer de préférence à n'importe quel moment et ainsi avoir un système qui s'ajuste de manière très rapide aux nouvelles préférences. Pour cela, des méthodes de désagrégation ont été mises au point.

Au sein de ce travail, nous avons proposé des algorithmes qui apportent une solution aux problématiques définies au sein de cette thèse. De plus, nous avons également proposé des algorithmes qui utilisent les informations issues des interactions entre les critères afin de mettre à jour les profils des utilisateurs.

Dans le cadre des expérimentations que nous avons menées pour valider et évaluer les performances de notre système, un travail complet a été réalisé afin de sélectionner les outils de méthodologie mis en œuvre lors des tests.

Nous avons effectué de nombreuses expérimentations. Elles ont été menées dans plusieurs cadres afin de tester notre système dans plusieurs conditions. Cela nous permet de pouvoir réellement valider notre système et de connaître les performances de nos algorithmes entre eux, mais également les performances de notre système vis-à-vis d'autres systèmes de recommandation.

Le test entre les performances de nos deux algorithmes a permis de montrer que notre algorithme Pairwise était meilleur que l'algorithme Rate.

Bien que l'échantillon soit limité, les tests réalisés avec de vrais utilisateurs utilisant directement notre système ont montré que notre système offrait de plutôt bonnes performances.

Les tests réalisés, avec de vrais utilisateurs n'utilisant pas directement notre système (utilisateurs de MovieLens), ou d'autres systèmes de recommandation (multi-agents, collaboratif, hybride), nous permettent de valider le fait que notre système soit efficace.

Cependant, il faut minorer ces résultats, en effet, la satisfaction de l'utilisateur sera fortement dépendante des résultats produits par le système de Recherche d'Information, et si ces résultats sont mauvais, alors l'utilisation de notre système ne pourra rien y changer. De plus, nous n'obtenons pas de bonnes performances quand de nombreux critères sont utilisés. Nous restons dépendants, de par le choix d'utiliser un système de recommandation basé sur le contenu, des informations disponibles à propos des éléments à recommander.

Dans la continuité de notre travail, nous avons défini plusieurs perspectives à explorer :

- Une première perspective consiste à déterminer de manière automatique, quel profil de l'utilisateur il est nécessaire d'utiliser, et ce, en fonction du contexte dans lequel l'utilisateur se trouve.
- Une seconde perspective serait d'utiliser des données contenues dans un profil au sein d'un autre, une sorte de système de profils collaboratif pour un même utilisateur. Cela permettrait d'obtenir des informations sur un profil que l'utilisateur n'a pas directement indiquées dans le contexte où il se trouve, mais où on peut retrouver l'information à partir d'un profil présentant des similarités.

- Un autre moyen que Kappalab afin d'utiliser les intégrales de Choquet pourrait être recherché afin de tester l'impact que ces intégrales peuvent avoir sur les performances de notre système.
- Une cinquième perspective serait d'utiliser une alternative aux intégrales de Choquet, en utilisant les CP-nets avec les réseaux GAI.
- Nous pourrions aussi tester si lors de l'utilisation des intégrales de Choquet, avec des valeurs u_{c_i} négatives, nous obtenons de meilleures performances. C'est-à-dire, si la possibilité, avec les intégrales de Choquet de pouvoir réduire la valeur d'un critère dans le profil de l'utilisateur, peut apporter de meilleures performances.
- Une sixième perspective à notre travail serait d'utiliser les scores présents dans les profils des utilisateurs afin de déterminer quels sont les vrais critères, quasi-critère et pseudo-critère. Cela permettant ainsi d'offrir une visualisation des préférences de l'utilisateur.
- Une septième perspective serait de comparer les performances de notre système avec celles de systèmes de recommandation autres que ceux avec lesquels nous nous sommes comparés.
- Une huitième perspective consiste à utiliser un système collaboratif en complément du système basé sur le contenu. Cela ne sera possible que lorsque de nombreux utilisateurs auront interagi avec notre système. Ainsi, l'on pourra exploiter les différents profils des utilisateurs afin de proposer de nouvelles recommandations. Il faudra donc mettre en place des algorithmes permettant d'exploiter de manière complémentaire les recommandations produites par les deux systèmes de recommandation. En effet, les systèmes de recommandations sont efficaces pour les utilisateurs ayant des goûts populaires. Ainsi une heuristique pourrait être définie pour utiliser le système collaboratif pour les utilisateurs aux goûts populaires, et le système basé sur le contenu pour les autres utilisateurs, mais aussi dans les cas où l'on veut recommander des objets peu évalués ou nouveaux.

Partie IV. Annexes

Mobifiz (mobifiz.fr)

Captures d'écran

Premièrement, voici la page d'accueil lorsque l'on est non connecté.

The screenshot shows the Mobifiz website homepage. At the top, there's a banner with the Mobifiz logo and the text 'JOUER AUX QUIZ GRATUITS ET GAGNER À COUP SÛR DE VRAIS CADEAUX !!!'. To the right is a 'Connexion' (Login) section with a username field, a password field with a 'se souvenir le mot de passe' checkbox, a 'Mot de passe perdu ?' link, and an 'Inscrivez-vous' (Sign up) button. Below the banner is a navigation bar with links: Accueil, Mon Compte, Créer / Gérer les quiz, Super Quiz, Quiz Solo, Gagnants du mois, and Parrainage.

The main content area is titled 'Bienvenue sur le quiz du jour !'. It features a cartoon character holding a book and the text 'Quiz du jour'. Below this, there's a section 'Affrontez vos amis sur Mobifiz !' with three difficulty level buttons: 'Impressionnisme', 'Peinture', and 'Facile'. To the right of these buttons is a box containing the text 'Tester vos connaissances sur cathédrale, 1894'. At the bottom of this section is a large orange button that says 'LANCEZ LE QUIZ'.

On the right side, there's a section 'A gagner !' (To win!) with a cartoon character and the text 'Mobifiz vous offre une place de cinéma, Inscrivez-vous vite !' and a 'Je participe !' button. Below this is a section '5 derniers quiz du jour' (Last 5 quizzes of the day) listing: 'Les îles et archipels...', 'Musique classique', 'Quiz Johnny Hallyday', 'Culture G Bandes dessinées...', and 'Les fruits et les arb...'. At the bottom of the page is a section 'Plus de quiz ?' (More quizzes?) with a grid of categories: Divertissements (BD, Cinéma et TV, Jeux, Musique), Histoire-Géo (Préhistoire, Moyen-Âge, Renaissance, Capitales), Arts (Sculpture, Poésie, Peinture, Littérature), Technologie (Biologie, Informatique, Inventions, Math et Physique), Culture générale (Culture G, Politique, Mythologies), Sport (Football, Rugby, Cyclisme, Auto-Moto), Nature (Dev. Durable, Faune, Flore), and Pour elle (People, Mode, Déco, Beauté). At the bottom center of this section is an 'Inscrivez-vous' button.

At the bottom left of the main content area, there's a 'FLASH INFO' section with a text box saying 'Prochain quiz à 9h, venez nombreux...'. To the right of this is a 'PARTAGEZ SUR' section with social media share buttons for Facebook, Twitter, and Google+.

Voici la page d'accueil lorsque l'on est connecté. On remarque que sur la droite de la page on trouve deux listes. Une liste de quiz sélectionné, et une liste de quiz recommandé. La première liste concerne une sélection des quiz les plus joués. Et la seconde est une liste générée en fonction des préférences de l'utilisateur en fonction de tous les quiz disponible sur la plateforme.

MOBIFIZ JOUE AUX QUIZ GRATUITS ET GAGNE A COUP SÛR DE VRAIS CADEAUX !!!

Connexion
 Bienvenue arnaud3013
 Déconnexion
 Votre grade : **1300** RESTANTS AUJOURD'HUI
5950 MON TOTAL

Accueil | Mon Compte | Créer / Gérer les quiz | Super Quiz | Quiz Solo | Gagnants du mois | Parrainage

Super Quiz

Des places de cinéma à gagner !!!
 Participez au SUPER QUIZ et gagnez des places de cinéma par tirage au sort

JOUER

fizeo
 Le partenaire de vos projets

FIZ TE FAIT GAGNER À COUP SÛR DE VRAIS CADEAUX !!!

MON COMPTE | PARRAINAGE | GAGNANTS DU MOIS | QUIZ SOLO | SUPER QUIZ | GÉRER LES QUIZ

Quiz sélectionnés

- Le quiz sur les perso...
- Beyoncé
- La série Plus belle l...
- Connaissances en poés...
- Connaissiez-vous les c...

Quiz recommandé

- Appareils mythiques
- Canaux, fleuves et mer...
- La planète en danger
- Quiz sur le tennis
- Les dictons des pays

Partenaires

fizeo
 Le partenaire de vos projets

- Habitat
- Services à la personne
- Finances et assurances
- Services aux entreprises

Mobifiz sur votre mobile !

Gagnant du mois
 Quiz TOP 10
 Parrainez vos amis
 Suivez-nous

Mobifiz sur votre mobile !
 Télécharger gratuitement notre application sur votre mobile
 Scanner le QR Code

Fizeo | Contact | Charte qualité | Conditions générales | Plan du site | Partenaires | Affiliation | FAQ | Recherche

Tableaux des résultats obtenus lors de l'expérimentation

- « Place » correspond à la place du quiz dans la liste présentée.
- « NBQuiz » correspond au nombre de quiz présent dans la liste présentée à l'utilisateur.
- « IDUser » correspond à l'id de l'utilisateur (nous avons remplacé le véritable id pour des raisons de confidentialité)
- « IDQuiz » (nous avons remplacé le véritable id pour des raisons de confidentialité)
- « IDGroup » correspond au groupe dans lequel est présent l'utilisateur.
- « Indice_De_Performance » correspond à l'indice de performance (formule 56).

Place	NBQuiz	IDUser	IDQuiz	IDGroup	Indice_De_Performance
4	5	User 1	x	1	0, 25
17	31	User 1	x	1	0, 466666667
1	8	User 1	x	1	1
3	5	User 2	x	1	0, 5
5	5	User 2	x	1	0
2	5	User 2	x	1	0, 75
4	6	User 2	x	1	0, 4
2	8	User 2	x	1	0, 857142857
2	8	User 2	x	1	0, 857142857
4	8	User 2	x	1	0, 571428571
2	5	User 2	x	1	0, 75
1	6	User 3	x	3	1
1	28	User 3	x	3	1
3	28	User 3	x	3	0, 925925926
10	88	User 3	x	3	0, 896551724
1	88	User 3	x	3	1
4	79	User 3	x	3	0, 961538462
7	73	User 3	x	3	0, 916666667
3	73	User 3	x	3	0, 972222222
6	88	User 3	x	3	0, 942528736
4	88	User 3	x	3	0, 965517241
1	5	User 3	x	3	1
1	5	User 3	x	3	1
1	5	User 4	x	2	1
7	12	User 4	x	2	0, 454545455

62	73	User 4	x	2	0, 152777778
69	73	User 4	x	2	0, 055555556
46	79	User 4	x	2	0, 423076923
18	73	User 4	x	2	0, 763888889
36	73	User 4	x	2	0, 513888889
51	73	User 4	x	2	0, 305555556
62	73	User 4	x	2	0, 152777778
61	79	User 4	x	2	0, 230769231
65	79	User 4	x	2	0, 179487179
4	10	User 4	x	2	0, 666666667
67	79	User 4	x	2	0, 153846154
3	10	User 4	x	2	0, 777777778
3	5	User 4	x	2	0, 5
2	10	User 4	x	2	0, 888888889
5	5	User 4	x	2	0
4	5	User 4	x	2	0, 25
6	10	User 4	x	2	0, 444444444
1	8	User 4	x	2	1
2	8	User 4	x	2	0, 857142857
4	8	User 4	x	2	0, 571428571
8	8	User 4	x	2	0
2	10	User 4	x	2	0, 888888889
3	10	User 4	x	2	0, 777777778
5	10	User 4	x	2	0, 555555556
7	10	User 4	x	2	0, 333333333
12	79	User 4	x	2	0, 858974359
41	79	User 4	x	2	0, 487179487
4	73	User 4	x	2	0, 958333333
2	12	User 4	x	2	0, 909090909
2	5	User 5	x	1	0, 75
4	7	User 5	x	1	0, 5
4	7	User 5	x	1	0, 5
5	7	User 5	x	1	0, 333333333
3	7	User 5	x	1	0, 666666667
2	6	User 5	x	1	0, 8
4	6	User 5	x	1	0, 4
6	6	User 5	x	1	0
3	6	User 5	x	1	0, 6
5	6	User 5	x	1	0, 2
38	45	User 5	x	1	0, 159090909
33	45	User 5	x	1	0, 272727273

18	45	User 5	x	1	0, 613636364
1	5	User 5	x	1	1
3	5	User 5	x	1	0, 5
4	5	User 5	x	1	0, 25
5	5	User 5	x	1	0
2	5	User 5	x	1	0, 75
5	5	User 5	x	1	0
4	25	User 6	x	2	0, 875
17	79	User 6	x	2	0, 794871795
3	5	User 7	x	3	0, 5
2	16	User 8	x	2	0, 933333333
3	8	User 8	x	2	0, 714285714
8	16	User 8	x	2	0, 533333333
3	5	User 8	x	2	0, 5
24	25	User 8	x	2	0, 041666667
1	25	User 8	x	2	1
3	5	User 8	x	2	0, 5
4	5	User 8	x	2	0, 25
1	26	User 8	x	2	1
1	5	User 8	x	2	1
1	5	User 9	x	3	1
2	31	User 9	x	3	0, 966666667
1	5	User 9	x	3	1
1	5	User 9	x	3	1
2	5	User 9	x	3	0, 75
1	5	User 9	x	3	1
4	5	User 9	x	3	0, 25
5	26	User 9	x	3	0, 84
1	6	User 10	x	1	1
4	6	User 10	x	1	0, 4
6	28	User 10	x	1	0, 814814815
2	28	User 10	x	1	0, 962962963
10	23	User 10	x	1	0, 590909091
3	8	User 11	x	2	0, 714285714
2	5	User 12	x	3	0, 75
4	5	User 12	x	3	0, 25
2	6	User 12	x	3	0, 8
2	5	User 12	x	3	0, 75
3	5	User 12	x	3	0, 5
5	5	User 12	x	3	0
1	45	User 13	x	3	1

4	5	User 14	x	1	0, 25
5	5	User 14	x	1	0
2	5	User 14	x	1	0, 75
3	5	User 14	x	1	0, 5
1	79	User 15	x	2	1
5	79	User 15	x	2	0, 948717949
14	79	User 15	x	2	0, 833333333
25	79	User 15	x	2	0, 692307692
41	79	User 15	x	2	0, 487179487
51	79	User 15	x	2	0, 358974359
70	79	User 15	x	2	0, 115384615
1	45	User 15	x	2	1
1	45	User 15	x	2	1
30	45	User 15	x	2	0, 340909091
38	45	User 15	x	2	0, 159090909
44	45	User 15	x	2	0, 022727273
4	73	User 15	x	2	0, 958333333
19	73	User 15	x	2	0, 75
29	47	User 15	x	2	0, 391304348
37	47	User 15	x	2	0, 217391304
2	5	User 15	x	2	0, 75
5	5	User 15	x	2	0
4	5	User 15	x	2	0, 25
3	5	User 15	x	2	0, 5
1	5	User 15	x	2	1
2	16	User 15	x	2	0, 933333333
8	16	User 15	x	2	0, 533333333
9	26	User 15	x	2	0, 68
2	26	User 15	x	2	0, 96
3	6	User 15	x	2	0, 6
3	16	User 15	x	2	0, 866666667
2	16	User 15	x	2	0, 933333333
7	16	User 15	x	2	0, 6
9	16	User 15	x	2	0, 466666667

Ainsi on obtient par utilisateur, les moyennes de performances suivantes :

IDUser	MoyPerf/user
User 1	0, 572222222
User 2	0, 585714286

User 3	0, 965079248
User 4	0, 519730717
User 5	0, 436602871
User 6	0, 834935897
User 7	0, 5
User 8	0, 647261905
User 9	0, 850833333
User 10	0, 753737374
User 11	0, 714285714
User 12	0, 578571429
User 13	1
User 14	0, 375
User 15	0, 611632901

Et par groupe, la moyenne de performance de chaque utilisateur :

MoyPerf/user	Groupe 1	Groupe 2	Groupe 3
	0, 572222222	0, 51973072	0, 965079248
	0, 58571429	0, 8349359	0, 5
	0, 43660287	0, 6472619	0, 850833333
	0, 75373737	0, 71428571	0, 578571429
	0, 375	0, 6116329	1

Et enfin la moyenne de performance par groupe :

Moyenne/Groupe	0, 54465535	0, 66556943	0, 778896802
----------------	-------------	-------------	--------------

Movielens

Méthode Rate sur la BDD restreinte

Moyenne de Kendall Tau-b	Valeur de Kendall Tau-B pour la K-partie = 1	Valeur de Kendall Tau-B pour la K-partie = 2	Valeur de Kendall Tau-B pour la K-partie = 3	Valeur de Kendall Tau-B pour la K-partie = 4	Valeur de Kendall Tau-B pour la K-partie = 5	Valeur de Kendall Tau-B pour la K-partie = 6
1	1	1	1	1	1	1
0	-0, 5	-0, 5	-0, 5	1	1	-0, 5
0, 20939153	0, 77777778	0	0, 42857143	0, 25	-0, 2	0
0, 05681818	-0, 5	-0, 45454546	0, 54545455	-0, 5	0, 25	1
0, 0111578	-0, 49019608	0, 08928571	0	0, 125	0, 2	0, 14285714
0, 10985306	-0, 16666667	0, 6	-0, 07142857	0	-0, 17647059	0, 47368421
0, 14826794	0, 09574468	-0, 01639344	-0, 04301075	0, 11363636	0, 34710744	0, 39252336
0, 08496481	0, 10435764	0, 16210204	0, 15533704	-0, 0503639	0, 02546936	0, 11288667
0, 26666667	0	-0, 33333333	0, 6	0, 33333333	0	1
0, 00920473	-0, 088	0, 19512195	0, 07692308	0, 00787402	-0, 01438849	-0, 12230216
-0, 05364373	0, 20588235	0, 03846154	-0, 20588235	-0, 25641026	-0, 05263158	-0, 05128205
0, 14838557	0, 04166667	0, 18518519	0, 1	-0, 1	0, 53846154	0, 125
0, 04288612	0, 02072539	-0, 20212766	-0, 03	0, 34517767	-0, 00534759	0, 12888889
-0, 00213863	0, 10891089	-0, 19	0, 28181818	-0, 12621359	-0, 11538462	0, 02803738
-0, 01696429	0, 16666667	-0, 5	0, 28571429	-0, 1875	-0, 16666667	0, 3
0, 15164818	0, 04761905	0, 375	0, 42105263	0, 1025641	0, 20689655	-0, 24324324
0,	0, 01558153	0, 00846979	0, 12939841	0, 13690816	0, 03037383	-0,

03725158						09722222
0, 0041208	0, 01414141	-0, 00982318	-0, 10309278	0, 28048781	-0, 21581197	0, 05882353
0, 08055556	0, 53333333	-0, 16666667	0	-0, 16666667	0, 41666667	-0, 13333333
-0, 01736111	-0, 375	0, 4375	-0, 16666667	-0, 0625	0, 1875	-0, 125
0, 04166667	-0, 375	0, 25	0, 25	0, 625	0, 25	-0, 75
-0, 01686508	-0, 57142857	-0, 375	0, 625	0, 42857143	-0, 375	0, 16666667
-0, 01893939	-0, 36363636	0, 25	-0, 33333333	0	0	0, 33333333
0, 36666667	1	-0, 4	-0, 4	0	1	1
0, 08434343	-0, 5	-0, 16666667	0, 3	0, 3	0, 3	0, 27272727
0, 02856676	-0, 11428571	0, 15483871	-0, 12820513	0, 02020202	0, 09677419	0, 1420765
-0, 12222222	0	0, 6	-0, 33333333	-0, 33333333	-0, 33333333	-0, 33333333
0, 05116842	0, 06846847	-0, 00172499	-0, 00127714	0, 10157195	0, 11135058	0, 02862164
-0, 05778604	0	-0, 22826087	0, 08854167	-0, 02162162	-0, 05319149	-0, 13218391
-0, 06989661	0, 02543941	-0, 04621658	-0, 04351809	-0, 19107981	-0, 05764761	-0, 10635697
0, 04872409	0, 07246377	0, 45454546	0, 21428571	0, 05970149	-0, 11428571	-0, 3943662
0, 03321648	-0, 0112	-0, 06983655	0, 16139241	-0, 1761194	0, 02887538	0, 26618705
0, 07341867	-0, 15384615	-0, 03571429	-0, 125	-0, 36666667	0, 52173913	0, 6
-0, 11357274	-0, 01843318	-0, 27753304	-0, 21585903	-0, 17040359	-0, 0195122	0, 02030457
0, 09082492	0, 25	0, 22222222	-0, 2	0, 18181818	-0, 18181818	0, 27272727
0, 10554538	0, 29268293	-0, 07317073	0, 28	0, 30357143	-0, 16981132	0
0, 25415445	-0, 30188679	0, 37254902	0, 33333333	0, 42857143	0, 11764706	0, 57471264
0,	-0, 1372549	0, 05882353	0, 10638298	0, 48	-0, 24489796	0,

14841744						62745098
0, 43333333	-0, 4	0	1	1	1	0
0, 11065369	0, 25	-0, 10569106	0, 10526316	0, 05072464	0, 29104478	0, 07258065
-0, 05555556	1	-1	-0, 33333333	-0, 5	-0, 5	1
0, 31771586	0, 46666667	0, 57142857	0, 05882353	0, 2631579	0, 11764706	0, 42857143
0, 0682664	0, 0139795	0, 3032967	-0, 15913556	-0, 03084416	0, 14098973	0, 14131219
0, 09654235	-0, 07692308	0, 18181818	0, 33333333	-0, 69230769	0, 33333333	0, 5
0, 15658089	0, 18446602	0, 3125	0, 41071429	0, 27522936	-0, 29292929	0, 04950495
-0, 29202783	-0, 04545455	-0, 43478261	-0, 33333333	-0, 33333333	-0, 10526316	-0, 5
-0, 10449735	-0, 22222222	-0, 25	0	0, 25	0, 16666667	-0, 57142857
0, 06099628	0, 0731539	-0, 06023102	0, 02679901	0, 28671329	0, 05801825	-0, 01847575
0, 00598566	0	-0, 09565217	0, 2	-0, 25454546	-0, 13592233	0, 3220339
0, 05574277	0, 00436681	0, 05928854	0, 07868853	-0, 17667845	0, 28571429	0, 08307692
-0, 08333333	-0, 5	-0, 5	1	-0, 5	1	-1
0, 02335189	-0, 104	0, 14018692	-0, 07017544	-0, 46153846	0, 125	0, 5106383
0, 26666667	1	-0, 4	0	1	0	0
0, 21111111	-0, 4	0	-0, 33333333	1	1	0
0, 21862859	0, 07692308	0, 25	0, 5	0, 25	-0, 18181818	0, 41666667
-0, 28262323	-0, 4	-0, 66666667	-0, 10526316	0	-0, 16666667	-0, 35714286
-0, 00673688	-0, 08472068	0, 03675635	-0, 23297755	0, 19032486	-0, 03087541	0, 08107117
0, 4945055	-0, 07142857	0, 53846154	0, 25	0, 25	1	1
-0,	-0,	0, 39473684	-0,	0, 04651163	0, 16666667	-0, 325

10192321	61538462		27906977			
0, 04564629	0, 01724138	0, 12345679	0, 2	-0, 03478261	-0, 05882353	0, 02678571
0, 19047619	-0, 35714286	-0, 07142857	0	0, 78571429	1	-0, 21428571
-0, 05709536	-0, 39726027	0, 19736842	0, 34722222	-0, 30666667	0, 18518519	-0, 36842105
0, 27777778	-0, 5	-0, 5	1	-0, 33333333	1	1
0, 27777778	1	-0, 33333333	1	1	-0, 5	-0, 5
-0, 02721216	0, 08152174	-0, 24561404	-0, 13407134	0, 02213542	0, 09298781	0, 01976744
0, 88888889	1	1	1	0, 33333333	1	1
0, 25709394	-0, 06060606	0, 21875	0, 65517241	0, 32258065	0, 36666667	0, 04
0, 20296353	0, 45	-0, 52631579	0, 60869565	0, 58823529	0, 30769231	-0, 21052632
-0, 04077564	-0, 16363636	-0, 06451613	0, 24637681	-0, 1369863	0, 12727273	-0, 25316456
0, 06055584	-0, 0618878	0, 12417994	0, 25428082	0, 01889401	-0, 02407996	0, 05194805
0, 02185292	-0, 27272727	-0, 17142857	-0, 3902439	1	-0, 03448276	0
0, 33712121	0, 25	1	0, 36363636	0, 25	0, 25	-0, 09090909
-0, 20801542	-0, 31034483	-0, 24137931	-0, 04166667	0, 05405405	-0, 4516129	-0, 25714286
0, 10955593	-0, 05882353	0, 53061225	-0, 09	0, 01980198	0, 03669725	0, 21904762
0, 10128729	0, 40625	-0, 02439024	0, 13333333	0, 24637681	-0, 12307692	-0, 03076923
0, 05555556	0, 4	0	-0, 4	-0, 33333333	-0, 33333333	1
-0, 12326681	-0, 11764706	0	-0, 70588235	0, 07142857	-0, 1875	0, 2
-0, 01639074	0, 04347826	-0, 31168831	-0, 01470588	0, 11111111	-0, 03921569	0, 11267606
0, 04354476	-0, 21052632	1	-0, 46153846	-0, 06666667	0	0
0,	0, 1395881	0, 0204918	0, 26855895	0, 24285714	0, 16666667	0,

19757383						34728034
0, 03861738	0	0, 14285714	0, 15789474	0, 38095238	-0, 45	0
0, 10547386	-0, 375	0	0, 25	0, 41666667	-0, 05882353	0, 4
-0, 12775767	-0, 44783715	-0, 07142857	-0, 12785388	-0, 07670455	-0, 16272189	0, 12
0, 0196599	-0, 30923695	0	0, 28571429	-0, 08646617	-0, 04961832	0, 27756654
0, 18981482	0, 22222222	0, 66666667	-0, 25	1	-0, 25	-0, 25
-0, 00015509	-0, 14159292	0	0, 34126984	-0, 06837607	0	-0, 13223141
-0, 34265734	-0, 36363636	0	-0, 69230769	-0, 18181818	-1	0, 18181818
0, 04034567	-0, 07164053	0, 05307498	0, 0907577	0, 02496715	-0, 03410025	0, 17901499
0, 2281746	0, 16666667	-0, 14285714	0, 25	0, 16666667	0, 5	0, 42857143
0, 13888889	-0, 5	-0, 5	0, 33333333	-0, 5	1	1
0, 57203445	0, 54545455	0, 25	1	0, 5	0, 44444444	0, 69230769
0, 04777095	-0, 26801802	0, 07823961	0, 06502242	0, 18779343	0, 15525114	0, 06833713

Méthode Pairwise sur la BDD restreinte

Moyenne de Kendall Tau-b	Valeur de Kendall Tau-B pour la K-partie = 1	Valeur de Kendall Tau-B pour la K-partie = 2	Valeur de Kendall Tau-B pour la K-partie = 3	Valeur de Kendall Tau-B pour la K-partie = 4	Valeur de Kendall Tau-B pour la K-partie = 5	Valeur de Kendall Tau-B pour la K-partie = 6
1	1	1	1	1	1	1
0, 5	-0, 5	1	-0, 5	1	1	1
0, 44920635	0, 66666667	0, 5	0, 42857143	0, 5	0, 6	0
0, 95454545	1	0, 72727273	1	1	1	1

0, 57848622	0, 125	0, 16666667	1	0, 79310345	0, 62857143	0, 75757576
0, 2058283	-0, 16666667	0, 6	-0, 07142857	0	0, 29411765	0, 57894737
0, 36333195	0, 28947368	-0, 23529412	0, 27027027	0, 625	0, 34482759	0, 88571429
0, 00877852	0, 24324324	-0, 17948718	0, 23529412	-0, 17948718	-0, 175	0, 10810811
0, 51111111	0	0, 66666667	0	1	0, 4	1
0, 28688527	-0, 68965517	0, 51515152	-0, 21428571	0, 78787879	0, 72222222	0, 6
0, 77383367	0, 37931034	0, 47058824	1	0, 79310345	1	1
0, 48549383	-0, 16666667	0, 2962963	0, 55	0, 4	1	0, 83333333
0, 3659149	0, 19444444	0, 4137931	0, 24242424	0, 68965517	0, 31034483	0, 34482759
-0, 02161527	-0, 02631579	-0, 26829268	-0, 62162162	0, 225	0, 46153846	0, 1
0, 05363757	0, 44444444	-0, 16666667	0, 28571429	-0, 375	-0, 16666667	0, 3
0, 39850603	-0, 42857143	-0, 25	0, 76315789	0, 64102564	0, 82758621	0, 83783784
0, 175376	0, 63888889	-0, 05714286	0, 63636364	-0, 36585366	0, 05714286	0, 14285714
0, 06294669	-0, 03030303	-0, 08571429	0, 25	-0, 47058824	0	0, 71428571
0, 44166667	0, 13333333	1	0, 26666667	0, 41666667	-0, 16666667	1
0, 94791667	1	0, 6875	1	1	1	1
0, 73809524	0	1	0, 42857143	1	1	1
0, 8422619	0, 42857143	0, 625	1	1	1	1
0, 40909091	0, 27272727	1	-0, 33333333	0, 54545455	0, 63636364	0, 33333333
0, 76666667	1	1	0, 6	0	1	1
0, 39166667	-0, 66666667	0, 41666667	0, 3	1	0, 3	1

0, 78434925	0, 58333333	0, 47058824	0, 65217391	1	1	1
0, 34444444	-1	0, 4	1	-0, 33333333	1	1
0, 15342251	-0, 4137931	0, 06060606	0, 35483871	0, 13793103	0, 66666667	0, 11428571
-0, 01425486	0, 14285714	0, 16216216	-0, 34210526	-0, 36585366	0, 10810811	0, 20930233
0, 30840462	-0, 11428571	0, 27586207	0, 4	0, 35135135	-0, 0625	1
0, 18567965	-0, 2	0, 61904762	-0, 17073171	0, 41463415	0, 23684211	0, 21428571
0, 30982504	0, 07894737	0, 05263158	0, 20588235	0, 59375	0, 23076923	0, 6969697
0, 48789284	0, 26923077	0, 46428571	-0, 125	0, 83333333	0, 65217391	0, 83333333
0, 12918669	0, 65217391	0, 18421053	0, 07894737	-0, 66666667	-0, 21428571	0, 74074074
1	1	1	1	1	1	1
0, 72030766	0, 91666667	0, 2173913	0, 60714286	0, 83870968	0, 74193548	1
0, 78378852	0, 47058824	0, 85714286	1	1	0, 375	1
0, 58113322	0, 16666667	0, 6875	0, 25	0, 74074074	0, 89189189	0, 75
0, 61111111	0	0	1	1	1	0, 66666667
0, 59929862	0, 36842105	0, 34375	0, 46875	0, 65625	0, 75862069	1
0, 77777778	1	-0, 33333333	1	1	1	1
0, 05052582	-0, 73333333	1	-0, 35294118	0, 36842105	0, 23529412	-0, 21428571
0, 77888125	0, 35294118	0, 60606061	0, 71428571	1	1	1
0, 52253302	0, 53846154	0, 54545455	0, 33333333	0, 38461538	0, 33333333	1
0, 32174672	0, 02777778	0, 45714286	0, 41025641	0, 43243243	0, 42105263	0, 18181818
0, 31711525	0	-0, 47826087	0, 71428571	-0, 33333333	1	1
0,	-0,	-0, 25	1	0, 5	1	1

50462963	22222222					
0, 64645969	0, 2	0, 69444444	0, 61764706	0, 36666667	1	1
0, 17397983	-0, 34782609	0	-0, 06451613	0, 5483871	0, 19354839	0, 71428571
0, 82850242	0, 66666667	1	0, 30434783	1	1	1
1	1	1	1	1	1	1
0, 04454221	0, 30555556	0, 21428571	0, 42857143	-0, 21212121	-0, 12121212	-0, 34782609
1	1	1	1	1	1	1
0, 66666667	-0, 4	1	1	0, 4	1	1
0, 39957265	0, 23076923	-0, 5	0, 5	0, 5	1	0, 66666667
-0, 0911825	-0, 36363636	-0, 4	-0, 52631579	-0, 4	1	0, 14285714
0, 25112576	-0, 37037037	0, 2173913	-0, 14814815	1	0, 37931034	0, 42857143
0, 68482906	0, 5	1	1	0, 25	0, 66666667	0, 69230769
-0, 2958503	-0, 60606061	-0, 03225806	-0, 4	-0, 19444444	-0, 35483871	-0, 1875
0, 37193818	0, 06060606	0, 08695652	0	0, 5483871	0, 83870968	0, 6969697
0, 36471861	0, 57142857	-0, 35714286	-0, 45454545	1	1	0, 42857143
0, 20170966	-0, 625	0, 36363636	-0, 025	0, 875	0, 2972973	0, 32432432
0, 88888889	1	1	1	0, 33333333	1	1
0, 88888889	1	0, 33333333	1	1	1	1
0, 75047438	0, 5483871	0, 48387097	1	1	0, 47058824	1
1	1	1	1	1	1	1
0, 44016396	0, 24242424	0, 375	0, 65517241	0, 5483871	0, 7	0, 12
0, 56201779	0, 47058824	0, 52631579	0, 52380952	1	0, 58823529	0, 26315789
0, 24748251	0, 04761905	-0, 13793103	-0, 11111111	0, 5483871	1	0, 13793103

0, 12180521	-0, 27272727	0, 42105263	0, 42105263	-0, 41935484	0, 27777778	0, 3030303
0, 67266419	0, 09677419	0, 85185185	0, 58333333	0, 85185185	1	0, 65217391
0, 56439394	-0, 5	1	0, 63636364	0, 25	1	1
0, 33866396	-0, 86206897	0, 37931034	0, 66666667	0, 51351351	0, 67741935	0, 65714286
0, 17110993	-0, 1	0, 35135135	-0, 08333333	0, 10526316	0, 37837838	0, 375
0, 32710116	-0, 23529412	0, 24137931	1	-0, 04347826	0	1
0, 44444444	-0, 33333333	0, 4	-0, 4	1	1	1
0, 06748561	-0, 05555556	-0, 5625	0, 06666667	0, 47058824	0, 28571429	0, 2
0, 28325888	0, 42857143	-0, 25	0, 2173913	0, 35294118	0, 31428571	0, 63636364
0, 31990553	-0, 31578947	0, 33333333	0, 46153846	0, 46666667	0, 47368421	0, 5
0, 08425156	-0, 57142857	0, 25641026	0, 4	-0, 42857143	0, 41666667	0, 43243243
0, 34415205	0	0, 19047619	0, 63157895	0, 14285714	0, 4	0, 7
0, 39593254	0, 1875	0, 25	0, 2	0, 3	0, 86666667	0, 57142857
0, 70467467	0, 60714286	0, 4	0, 53125	0, 68965517	1	1
0, 15027507	-0, 4	0, 08333333	-0, 09090909	0, 58333333	0, 25714286	0, 46875
0, 23148148	0, 22222222	0	1	0, 16666667	-0, 25	0, 25
0, 14067173	0	-0, 57142857	0, 25	0, 30434783	0, 58333333	0, 27777778
0, 02651515	-0, 81818182	0, 09090909	0	-0, 36363636	1	0, 25
0, 29699853	0, 20689655	0, 37931034	0, 06896552	-0, 07407407	0, 85714286	0, 34375
0, 9047619	1	0, 42857143	1	1	1	1
0, 88888889	0, 33333333	1	1	1	1	1

0, 46969697	-0, 18181818	-0, 5	1	0, 5	1	1
-0, 12061488	-0, 225	-0, 07317073	-0, 20930233	0	0, 16216216	-0, 37837838
0, 68166441	0, 44444444	0, 67741935	0, 52380952	0, 8125	0, 83870968	0, 79310345
0, 43333333	0, 4	-0, 4	0	1	0, 6	1
0, 48160161	-0, 375	1	0, 48387097	0, 33333333	0, 83870968	0, 60869565
0, 88707335	1	0, 85185185	0, 47058824	1	1	1
0, 84444444	0, 66666667	1	1	1	1	0, 4
0, 06014196	-0, 21212121	-0, 02702703	0, 39473684	0, 10526316	0, 1	0
0, 22598773	-0, 4	-0, 21875	0, 35135135	0, 42857143	0, 45945946	0, 73529412
0, 63333333	0, 4	0	1	1	1	0, 4

Méthode Pairwise sur la BDD riche

Moyenne de Kendall Tau-b	Valeur de Kendall Tau-B pour la K-partie = 1	Valeur de Kendall Tau-B pour la K-partie = 2	Valeur de Kendall Tau-B pour la K-partie = 3	Valeur de Kendall Tau-B pour la K-partie = 4	Valeur de Kendall Tau-B pour la K-partie = 5	Valeur de Kendall Tau-B pour la K-partie = 6
1	1	1	1	1	1	1
0, 25	-0, 5	-0, 5	1	1	1	-0, 5
0, 4760582	0, 77777778	0, 25	0, 42857143	1	0, 4	0
0, 50378788	-0, 5	0, 54545455	0, 72727273	1	0, 25	1
0, 08409946	-0, 21875	-0, 25	0, 60714286	0, 20689655	0, 37142857	-0, 21212121
0, 02974286	-0, 16666667	0, 2	0	-0, 375	-0, 05882353	0, 57894737
-0, 11988985	-0, 28947368	-0, 23529412	-0, 05405405	-0, 375	0, 03448276	0, 2
0,	0,	-0,	-0,	0,	-0, 15	-0,

04084865	54054054	02564103	05882353	12820513		18918919
0, 54444444	0	1	0, 6	1	1	-0, 33333333
0, 06447107	-0, 20689655	0, 15151515	-0, 42857143	0, 36363636	0, 25	0, 25714286
0, 46339664	0, 27586207	-0, 23529412	0, 68965517	0, 4137931	1	0, 63636364
0, 05713438	0, 33333333	-0, 2962963	-0, 35	0, 05	0, 73076923	-0, 125
-0, 08445083	-0, 13888889	-0, 31034483	-0, 33333333	0, 55172414	-0, 17241379	-0, 10344828
0, 1004508	-0, 15789474	0, 02439024	0, 08108108	0, 45	0, 20512821	0
0, 0541336	0, 66666667	-0, 05555556	-0, 35714286	-0, 0625	-0, 16666667	0, 3
0, 03841052	-0, 42857143	-0, 25	0, 26315789	0, 05128205	0	0, 59459459
-0, 0444943	0, 05555556	0	-0, 36363636	-0, 07317073	0, 25714286	-0, 14285714
0, 09004744	0, 33333333	-0, 05714286	0, 22222222	0	-0, 17241379	0, 21428571
0, 1	0	-0, 16666667	0, 53333333	0, 41666667	0, 41666667	-0, 6
0, 40972222	0, 375	0, 6875	-0, 16666667	0, 375	0, 4375	0, 75
0, 37003968	-0, 375	0, 16666667	1	1	0	0, 42857143
0, 53075397	0, 14285714	0, 625	0, 25	1	1	0, 16666667
0, 01136364	-0, 36363636	0, 25	0, 33333333	0, 27272727	-0, 09090909	-0, 33333333
0, 76666667	1	0	0, 6	1	1	1
0, 13198653	-0, 27777778	-0, 16666667	0, 3	1	0, 3	-0, 36363636
0, 14257332	0, 16666667	0, 35294118	0, 13043478	0, 2173913	0, 37931034	-0, 39130435
0, 22222222	0	0	1	-0, 33333333	-0, 33333333	1
0, 25055807	0, 06896552	0, 39393939	0, 35483871	-0, 24137931	0, 55555556	0, 37142857
0,	0	0,	-0,	0, 2195122	0,	0,

10454163		40540541	34210526		13513514	20930233
0, 1854684	-0, 2	0, 31034483	0, 57142857	0, 18918919	-0, 0625	0, 30434783
0, 19294527	-0, 35	0, 4047619	-0, 07317073	0, 65853659	0, 18421053	0, 33333333
0, 23137589	0, 44736842	0, 07894737	-0, 08823529	0, 125	0, 12820513	0, 6969697
0, 09149745	-0, 30769231	0, 46428571	-0, 125	-0, 13333333	0, 2173913	0, 43333333
0, 07770382	0, 65217391	-0, 18421053	0, 07894737	-0, 25	-0, 57142857	0, 74074074
0, 93939394	1	1	1	0, 63636364	1	1
0, 46534268	0, 83333333	0, 60869565	0, 60714286	-0, 06451613	0, 48387097	0, 32352941
0, 37859866	0, 35294118	0, 60714286	-0, 11111111	0, 04761905	0, 375	1
0, 14952974	-0, 19444444	0, 375	0, 5	-0, 37037037	0, 24324324	0, 34375
0, 1	0	0	0	1	-0, 4	0
0, 37630305	0, 42105263	0, 34375	-0, 15625	0, 625	0, 17241379	0, 85185185
0, 30555556	-0, 5	-1	0, 33333333	1	1	1
0, 425559	0, 46666667	0, 57142857	0, 11764706	0, 26315789	0, 70588235	0, 42857143
0, 47706583	0, 55882353	0, 33333333	0, 42857143	0, 375	1	0, 16666667
-0, 37354312	-0, 23076923	0, 18181818	-1	-0, 69230769	-1	0, 5
-0, 09453767	-0, 16666667	-0, 31428571	0, 41025641	0, 16216216	-0, 05263158	-0, 60606061
0, 1224537	-0, 31818182	-0, 17391304	0, 42857143	-0, 33333333	0, 63157895	0, 5
0, 3287037	-0, 44444444	-0, 25	1	0, 5	0, 16666667	1
0, 16802832	-0, 3	0, 47222222	-0, 35294118	0, 3	1	-0, 11111111
0, 10911307	-0, 04347826	0, 21428571	-0, 06451613	0, 5483871	0	0
0, 3398752	0, 375	1	0, 30434783	0, 30434783	-0, 11111111	0, 16666667

0, 63888889	1	-0, 5	1	1	0, 33333333	1
-0, 03782912	0, 36111111	0, 21428571	0	-0, 33333333	-0, 12121212	-0, 34782609
0, 3	0, 4	0, 4	0	1	0	0
0, 73333333	0	1	1	0, 4	1	1
0, 06585082	0, 07692308	-0, 5	0, 25	0, 5	-0, 18181818	0, 25
-0, 03855092	-0, 36363636	-0, 4	-0, 21052632	-0, 4	1	0, 14285714
-0, 0110179	0, 14814815	-0, 34782609	-0, 37037037	0, 6	-0, 31034483	0, 21428571
0, 27228327	-0, 07142857	0, 61538462	0, 25	1	-0, 08333333	-0, 07692308
-0, 13210441	-0, 3030303	-0, 03225806	-0, 37142857	0, 13888889	-0, 19354839	-0, 03125
0, 15032513	0, 12121212	-0, 39130435	0, 33333333	0, 38709677	0, 4516129	0
0, 10714286	0, 64285714	-0, 07142857	-0, 45454545	0, 28571429	0, 45454545	-0, 21428571
0, 09048082	-0, 275	-0, 39393939	0, 075	0, 8125	-0, 10810811	0, 43243243
0, 77777778	1	1	1	-0, 33333333	1	1
0, 66666667	1	-1	1	1	1	1
0, 1753014	0, 19354839	0, 16129032	1	0, 36363636	0	-0, 66666667
1	1	1	1	1	1	1
0, 16721878	-0, 24242424	0, 3125	0, 65517241	0, 25806452	0, 1	-0, 08
0, 35205907	0, 35294118	-0, 31578947	0, 52380952	0, 7	0, 58823529	0, 26315789
0, 22353615	0, 04761905	0, 27586207	-0, 11111111	0, 5483871	0, 16666667	0, 4137931
0, 11154128	0, 06060606	0, 52631579	0, 31578947	0, 06451613	-0, 38888889	0, 09090909
0, 34513034	-0, 41935484	0, 85185185	0, 16666667	0, 44444444	0, 375	0, 65217391
0, 41287879	0, 25	1	0, 36363636	0, 25	0, 25	0, 36363636

0, 20693075	-0, 31034483	-0, 24137931	0, 375	0, 40540541	0, 61290323	0, 4
0, 08484471	-0, 4	0, 10810811	0	0, 28947368	0, 48648649	0, 025
0, 11145204	-0, 23529412	0, 34482759	-0, 11111111	0, 08695652	0, 25	0, 33333333
0, 3	1	-0, 8	-0, 4	1	0	1
0, 09016301	-0, 22222222	-0, 1875	-0, 13333333	-0, 05882353	0, 14285714	1
0, 03596939	0, 28571429	-0, 25	-0, 04347826	-0, 23529412	0, 42857143	0, 03030303
0, 15762483	-0, 31578947	0, 33333333	0, 46153846	0, 46666667	0	0
-0, 21186214	-0, 57142857	0, 12820513	-0, 48571429	-0, 42857143	0, 19444444	-0, 10810811
-0, 1781746	-0, 19047619	-0, 14285714	0	-0, 28571429	0	-0, 45
-0, 00297619	0, 125	0	-0, 2	0, 3	0, 4	-0, 64285714
0, 32522065	0, 53571429	0, 37142857	-0, 46875	0, 13793103	0, 375	1
0, 20118597	-0, 02857143	0, 16666667	0, 18181818	0, 16666667	0, 31428571	0, 40625
0, 30555556	0, 44444444	0, 22222222	1	0, 16666667	-0, 25	0, 25
-0, 3617629	-0, 37037037	-0, 57142857	-0, 21428571	-0, 34782609	-0, 66666667	0
-0, 31060606	-0, 45454545	-0, 18181818	0	-0, 72727273	0	-0, 5
0, 17737506	0, 20689655	0, 27586207	0, 24137931	0, 14814815	-0, 21428571	0, 40625
-0, 03075397	-0, 66666667	-0, 57142857	-0, 375	1	0	0, 42857143
0, 77777778	-0, 33333333	1	1	1	1	1
0, 49134199	0, 09090909	-0, 5	1	0, 35714286	1	1
-0, 01736103	-0, 325	0, 04878049	-0, 06976744	0, 05263158	0, 37837838	-0, 18918919
0, 37117447	0, 44444444	0, 38709677	0, 04761905	0, 25	0, 58064516	0, 51724138
0,	0, 4	0, 4	0	0, 4	1	1

53333333						
0, 47198955	0, 20833333	1	0, 74193548	0, 14814815	0, 51612903	0, 2173913
0, 82175926	0, 375	0, 55555556	1	1	1	1
0, 67777778	0, 66666667	0	1	1	1	0, 4
-0, 05938115	0, 3030303	-0, 05405405	0, 05263158	-0, 15789474	-0, 5	0
0, 04544794	-0, 11428571	-0, 21875	0, 35135135	0	-0, 21621622	0, 47058824
0, 4	-0, 4	0, 4	0, 4	1	0, 6	0, 4

Système Multi-Agents

Résultats sur BDD restreinte

Moyenne de Kendall Tau-b	Valeur de Kendall Tau-B pour la K-partie = 1	Valeur de Kendall Tau-B pour la K-partie = 2	Valeur de Kendall Tau-B pour la K-partie = 3	Valeur de Kendall Tau-B pour la K-partie = 4	Valeur de Kendall Tau-B pour la K-partie = 5	Valeur de Kendall Tau-B pour la K-partie = 6
1	1	1	1	1	1	1
0, 25	1	-0, 5	-0, 5	1	1	-0, 5
0, 46349206	0	0, 5	0, 71428571	0, 5	0, 4	0, 66666667
0, 59848485	0, 25	0, 54545455	0, 54545455	0, 25	1	1
0, 02750214	0, 01960784	-0, 17857143	0, 23728814	0, 375	-0, 14545455	-0, 14285714
0, 11024897	-0, 16666667	-0, 13333333	0	0, 1875	0, 35294118	0, 42105263
-0, 02608116	-0, 11702128	0, 31147541	-0, 21505376	-0, 21212121	-0, 08264463	0, 15887851
0, 10612585	0, 02002538	0, 17086056	0, 06854781	0, 17845706	0, 05093873	0, 14792552
0, 26666667	1	-0, 66666667	1	-0, 33333333	-0, 4	1
0, 08225254	0, 056	0, 02439024	0, 01398601	0, 45669291	0, 11510791	-0, 17266187

0, 06608716	-0, 05882353	0, 15384615	-0, 05882353	0, 1025641	0, 05263158	0, 20512821
0, 28631292	0, 5	0, 18518519	0, 55	0, 05	0, 30769231	0, 125
0, 09614771	0, 12435233	-0, 04255319	-0, 185	0, 36548223	0, 31016043	0, 00444444
0, 17125881	0, 03960396	0, 09	0, 27272727	0, 11650485	0, 14423077	0, 36448598
-0, 27202381	-0, 16666667	-0, 16666667	-0, 35714286	-0, 375	-0, 16666667	-0, 4
0, 21905452	0, 52380952	1	0, 42105263	-0, 12820513	-0, 44827586	-0, 05405405
0, 24536552	0, 22036728	0, 20158103	0, 1276958	0, 30576155	0, 31016355	0, 30662393
0, 01873991	0, 04646465	0, 07465619	0, 09896907	-0, 13821138	0, 06837607	-0, 03781513
0, 25277778	-0, 06666667	0, 41666667	-0, 06666667	0, 41666667	0, 41666667	0, 4
-0, 11111111	-0, 1875	0, 1875	-0, 16666667	-0, 3125	0	-0, 1875
-0, 125	0, 5	0, 25	-0, 375	-0, 75	-0, 375	0
0, 37301587	0, 14285714	1	0, 5	0, 42857143	0	0, 16666667
0, 30429293	0, 27272727	0, 25	1	1	-0, 36363636	-0, 33333333
0, 2	1	-0, 8	0	-1	1	1
0, 3212963	-0, 38888889	0, 41666667	0, 3	1	-0, 4	1
0, 11347286	0, 02857143	-0, 04516129	0, 17948718	0, 24242424	0, 19354839	0, 08196721
0, 1	0	0, 6	-0, 33333333	1	-0, 33333333	-0, 33333333
0, 06562432	0, 16525097	-0, 06727452	-0, 0293742	0, 09939541	-0, 00598659	0, 23173487
0, 09132765	-0, 00555556	0, 20652174	-0, 19270833	0, 03243243	0, 40957447	0, 09770115
0, 0072561	-0, 04856614	-0, 14318079	0, 12963811	-0, 11596244	0, 20734542	0, 01426243
-0, 04086281	-0, 10144928	0, 25757576	-0, 1	-0, 37313433	0, 1	-0, 02816901
0, 0791374	0, 2112	0, 19316493	-0, 2278481	-0, 01343284	0, 23404255	0, 07769784

0, 16410986	0, 80769231	-0, 03571429	-0, 125	-0, 26666667	0, 30434783	0, 3
0, 15335812	0, 39631336	-0, 04845815	0, 17180617	0, 39461883	-0, 0195122	0, 02538071
-0, 17180135	0, 25	-0, 44444444	-0, 2	0, 18181818	-0, 45454546	-0, 36363636
-0, 04650393	-0, 17073171	0, 26829268	-0, 06	-0, 23214286	-0, 01886793	-0, 06557377
0, 09931251	0, 28301887	0, 03921569	0	-0, 14285714	0, 11764706	0, 29885058
0, 0680647	-0, 1372549	0, 19607843	0, 21276596	0, 14	-0, 08163265	0, 07843137
0, 36666667	-0, 4	0, 33333333	1	1	-0, 4	0, 66666667
-0, 06050125	0, 00694444	-0, 05691057	0, 14035088	0, 04347826	-0, 23880597	-0, 25806452
-0, 08333333	1	0, 33333333	-0, 33333333	-0, 5	-0, 5	-0, 5
0, 10759497	0, 46666667	-0, 64285714	0, 11764706	0, 15789474	0, 11764706	0, 42857143
0, 0491789	0, 04380242	0, 05604396	0, 00098232	-0, 0413961	0, 0690943	0, 1665465
0, 05594406	-0, 07692308	0, 18181818	-0, 33333333	0, 23076923	0, 33333333	0
0, 08007495	-0, 13592233	0, 41964286	0, 07142857	0, 21100917	0, 17171717	-0, 25742574
0, 21375491	0, 59090909	-0, 13043478	0, 52380952	-0, 33333333	0, 63157895	0
0, 31679894	0, 22222222	-0, 25	0	0, 5	1	0, 42857143
0, 17855165	0, 27329193	-0, 02145215	0, 17766749	0, 21614749	0, 13754889	0, 28810624
0, 02858835	-0, 01041667	-0, 27826087	0, 49090909	0, 14545455	-0, 25242718	0, 07627119
-0, 08802019	-0, 01746725	-0, 07509881	-0, 09508197	-0, 25441696	0, 0031746	-0, 08923077
0, 77777778	1	1	1	1	1	-0, 33333333
0, 05190678	0, 064	0	0, 24561404	-0, 11538462	-0, 22321429	0, 34042553
0, 1	-0, 4	0	0	1	0	0
0, 1	-1	0	1	-0, 4	1	0

0, 00874126	-0, 53846154	-0, 5	0, 5	0, 5	0, 09090909	0
-0, 03132832	0, 3	-0, 13333333	0, 52631579	0	-0, 16666667	-0, 71428571
0, 04104643	0, 11024787	-0, 15233043	0, 05336769	0, 03001412	0, 0824555	0, 12252384
0, 17460318	0, 21428571	0, 23076923	0, 25	0, 25	0, 33333333	-0, 23076923
0, 15667598	0, 15384615	-0, 10526316	0, 41860465	0, 13953488	0, 33333333	0
0, 09849349	0, 25862069	0, 44444444	0, 13333333	0, 1826087	-0, 1512605	-0, 27678571
0, 15476191	0	0	0	0, 57142857	1	-0, 64285714
-0, 04760688	-0, 1369863	0, 21052632	0, 02777778	-0, 18666667	-0, 05555556	-0, 14473684
-0, 08333333	-0, 5	-0, 5	1	-1	1	-0, 5
0, 13888889	1	0, 33333333	-0, 5	-0, 5	-0, 5	1
0, 08775955	0, 25	-0, 03238866	0, 08118081	0	0, 12195122	0, 10581395
0, 5	1	1	1	1	-0, 5	-0, 5
0, 14523176	0, 12121212	0, 28125	-0, 44827586	0, 48387097	0, 03333333	0, 4
-0, 02963059	-0, 4	-0, 52631579	0, 39130435	0, 47058824	0, 30769231	-0, 42105263
0, 05338222	0, 12727273	0	-0, 02898551	0, 4109589	-0, 16363636	-0, 02531646
0, 04867105	-0, 04585931	0, 20196814	0, 11643836	-0, 08156682	0, 02544298	0, 07560297
-0, 07123241	-0, 45454546	-0, 17142857	0, 12195122	0, 11111111	-0, 03448276	0
0, 21212121	0, 25	1	0, 36363636	0, 25	-0, 5	-0, 09090909
0, 08278045	0, 13793103	0, 31034483	-0, 04166667	-0, 18918919	0, 19354839	0, 08571429
0, 06363278	0, 1372549	0, 48979592	0, 06	-0, 01980198	-0, 25688073	-0, 02857143
-0, 05135876	-0, 1875	0, 31707317	-0, 65333333	0, 24637681	0, 09230769	-0, 12307692
-0,	0, 4	0	-1	-0,	1	-0,

04444444				33333333		33333333
0, 14655696	0, 23529412	0	0	0, 28571429	-0, 375	0, 73333333
0, 21752294	0, 4673913	-0, 05194805	0, 33823529	0, 27777778	-0, 07843137	0, 35211268
-0, 01639676	-0, 21052632	-0, 33333333	-0, 46153846	-0, 06666667	0, 47368421	0, 5
-0, 01303619	-0, 06636156	0, 05532787	0, 05240175	-0, 05306122	0, 07364341	-0, 14016736
0, 06800334	0, 42857143	0, 28571429	-0, 31578947	-0, 19047619	0, 45	-0, 25
0, 03496732	0, 1875	-0, 4375	0, 25	-0, 16666667	0, 17647059	0, 2
-0, 0625309	-0, 22900763	-0, 00324675	-0, 12785388	-0, 03409091	0, 16568047	-0, 14666667
-0, 00942242	-0, 00401606	-0, 13170732	0, 14718615	-0, 36842105	0, 01145038	0, 28897338
-0, 31018519	0	-0, 44444444	-0, 25	-0, 66666667	-0, 25	-0, 25
-0, 0086415	-0, 20353982	0, 26213592	-0, 07936508	0, 05982906	0	-0, 09090909
0, 06228956	0	0, 27272727	0	0, 72727273	-0, 44444444	-0, 18181818
0, 01294312	0, 01441289	-0, 04844145	0, 08368027	-0, 03898379	-0, 01437962	0, 08137045
-0, 2202381	0, 16666667	-0, 57142857	0, 5	-0, 66666667	-0, 75	0
-0, 36111111	-0, 5	-0, 5	0, 33333333	-0, 5	-0, 5	-0, 5
0, 11486662	0, 09090909	0, 25	0, 25	-0, 5	0, 44444444	0, 15384615
0, 12266757	0, 14189189	0, 17848411	0, 09865471	0, 157277	0, 10730594	0, 0523918
0, 28497969	0, 06451613	0, 58823529	0, 47058824	0, 46153846	0	0, 125
0, 21111111	0, 4	0, 4	0, 4	0	0, 4	-0, 33333333
-0, 0563168	0, 11684783	-0, 07803468	-0, 18954248	-0, 0046083	-0, 16449086	-0, 01807229
0, 1655856	0, 2745098	-0, 03225807	0, 25179856	-0, 17490494	0, 27436823	0, 4
0,	1	0	1	0	0	0

33333333						
0, 00378717	0, 05376344	0, 43	-0, 28235294	-0, 22222222	0, 10526316	-0, 0617284
-0, 24580665	0	-0, 51923077	-0, 12	-0, 28301887	-0, 30769231	-0, 24489796
0, 23333333	-0, 4	-1	0, 4	1	1	0, 4

Résultats sur BDD riche

Moyenne de Kendall Tau-b	Valeur de Kendall Tau-B pour la K-partie = 1	Valeur de Kendall Tau-B pour la K-partie = 2	Valeur de Kendall Tau-B pour la K-partie = 3	Valeur de Kendall Tau-B pour la K-partie = 4	Valeur de Kendall Tau-B pour la K-partie = 5	Valeur de Kendall Tau-B pour la K-partie = 6
1	1	1	1	1	1	1
0, 5	-0, 5	1	-0, 5	1	1	1
0, 4218254	0	0, 5	0, 71428571	0, 25	0, 4	0, 66666667
0, 1969697	0, 25	0, 09090909	0, 09090909	-0, 5	0, 25	1
-0, 02474851	0, 23529412	-0, 17857143	0, 15254237	-0, 06944444	-0, 14545455	-0, 14285714
0, 13068241	-0, 16666667	0, 06666667	0	0, 1875	0, 11764706	0, 57894737
0, 02928896	0, 21276596	0, 04918033	-0, 04301075	-0, 03030303	-0, 11570248	0, 10280374
0, 12365077	-0, 00014102	0, 14847769	0, 09790509	0, 30029112	0, 06738466	0, 1279871
0, 08888889	-0, 4	0	1	-0, 66666667	-0, 4	1
0, 00987559	-0, 152	0, 04065041	-0, 02097902	0, 38582677	0, 11510791	-0, 30935252
0, 06746977	0, 20588235	0, 15384615	-0, 05882353	-0, 05128205	0, 05263158	0, 1025641
0, 29313865	-0, 25	0, 37037037	0, 55	0, 05	0, 53846154	0, 5
0, 11515522	0, 11398964	-0, 04255319	-0, 15	0, 54822335	0, 31016043	-0, 08888889
0, 20801075	-0, 05940594	0, 14	0, 12727273	0, 33009709	0, 32692308	0, 38317757

-0, 3280754	-0, 16666667	-0, 33333333	-0, 71428571	-0, 1875	-0, 16666667	-0, 4
0, 18783706	0, 04761905	1	0, 42105263	-0, 17948718	0	-0, 16216216
0, 18863805	0, 23539232	0, 18690006	0, 09194098	0, 23103252	0, 21028037	0, 17628205
-0, 01743854	0, 0040404	0, 08251474	0, 07628866	-0, 24186992	-0, 02350427	-0, 00210084
0, 04444444	-0, 26666667	0, 41666667	-0, 26666667	0, 41666667	-0, 16666667	0, 13333333
-0, 03819444	0, 1875	0, 375	-0, 16666667	-0, 0625	-0, 375	-0, 1875
-0, 20833333	0	0, 25	-0, 375	-0, 75	-0, 375	0
0, 37301587	0, 14285714	1	0, 5	0, 42857143	0	0, 16666667
0, 24368687	0, 27272727	0, 25	1	0, 63636364	-0, 36363636	-0, 33333333
0, 36666667	1	-0, 4	-0, 4	0	1	1
0, 36001684	0, 38888889	0, 41666667	0, 3	0, 3	0, 3	0, 45454546
0, 05363191	0, 02857143	0, 06451613	-0, 06410256	-0, 03030303	0, 29032258	0, 03278689
0, 1	0	0, 6	-0, 33333333	1	-0, 33333333	-0, 33333333
0, 1230074	0, 08262548	0, 05125678	0, 03422733	0, 13639661	0, 12547893	0, 30805925
0, 07740735	-0, 03888889	0, 16847826	-0, 27604167	0, 01081081	0, 39893617	0, 20114943
0, 00529727	-0, 12303423	0, 09198006	0, 01053596	-0, 08732394	0, 18363552	-0, 04400978
0, 06277148	-0, 14492754	0, 3030303	0, 02857143	0, 08955224	0, 12857143	-0, 02816901
0, 04940172	0, 192	0, 07429421	-0, 1835443	0, 02985075	0, 18237082	0, 00143885
0, 08172413	0, 23076923	-0, 03571429	-0, 125	0, 1	0, 08695652	0, 23333333
0, 13195936	0, 15207373	0	0, 2907489	0, 39461883	0	-0, 04568528
-0, 26271044	0, 25	-0, 44444444	-0, 2	0	-0, 45454546	-0, 72727273

-0, 05876311	-0, 17073171	0, 26829268	-0, 06	-0, 17857143	-0, 11320755	-0, 09836066
0, 04310633	0, 03773585	0, 03921569	0	-0, 14285714	0, 11764706	0, 20689655
0, 0839888	0, 03921569	0, 2745098	-0, 19148936	0, 14	0, 16326531	0, 07843137
0, 07777778	-0, 8	0, 66666667	0	1	-0, 4	0
-0, 00585908	-0, 03472222	0, 01626016	0, 18421053	0, 24637681	-0, 25373134	-0, 19354839
-0, 33333333	-0, 5	0, 33333333	-0, 33333333	-0, 5	-0, 5	-0, 5
-0, 06288024	0, 13333333	-0, 07142857	-0, 17647059	-0, 05263158	-0, 35294118	0, 14285714
0, 07631867	0, 05778192	-0, 06593407	-0, 03732809	0, 1948052	0, 18674136	0, 12184571
0, 26427739	0, 38461539	0, 18181818	-0, 33333333	0, 76923077	0, 33333333	0, 25
0, 06903413	-0, 12621359	0, 3125	0, 00892857	0, 18348624	0, 29292929	-0, 25742574
0, 07428155	0, 36363636	0, 2173913	-0, 04761905	0, 33333333	-0, 42105263	0
0, 44907407	0, 44444444	-0, 25	0	0, 5	1	1
0, 20086888	0, 25603865	0, 10231023	0, 27245658	0, 18944692	0, 16036506	0, 22459584
0, 07294956	-0, 02083333	0	0, 2	0, 31818182	-0, 13592233	0, 07627119
-0, 00624227	-0, 01746725	0, 05928854	-0, 09508197	-0, 00353357	0, 02857143	-0, 00923077
0, 41666667	1	1	1	-0, 5	0, 33333333	-0, 33333333
0, 01322832	0, 088	-0, 1682243	0, 38596491	-0, 00961539	-0, 3125	0, 09574468
-0, 16666667	-1	0	-1	1	0	0
0, 01111111	-1	0	-0, 33333333	0, 4	1	0
0, 13257576	0	0, 25	0, 08333333	0, 5	-0, 45454546	0, 41666667
-0, 0091061	0, 3	0	0, 52631579	0	-0, 16666667	-0, 71428571

0, 05529509	0, 11468738	-0, 20462296	0, 03643725	0, 09286723	0, 1449328	0, 14746882
0, 15033578	0, 35714286	0, 07692308	-0, 5	0, 25	0, 33333333	0, 38461539
0, 16073088	0	0, 2631579	0, 41860465	0, 04651163	0, 11111111	0, 125
-0, 00222292	0, 05172414	0, 12345679	-0, 025	-0, 05217391	-0, 07563025	-0, 03571429
0, 10281385	-0, 07142857	0	-0, 45454546	0, 57142857	1	-0, 42857143
-0, 01162261	-0, 08219178	0	0, 13888889	-0, 05333333	0, 11111111	-0, 18421053
-0, 08333333	-0, 5	-0, 5	1	-1	1	-0, 5
-0, 11111111	1	0, 33333333	-0, 5	-0, 5	-0, 5	-0, 5
0, 13605602	0, 30027174	0, 08502024	0, 16113161	0, 13802083	-0, 05182927	0, 18372093
0, 25	1	1	-0, 5	1	-0, 5	-0, 5
0, 23529204	0, 06060606	0, 53125	-0, 44827586	0, 35483871	0, 43333333	0, 48
0, 03010655	-0, 2	0	-0, 13043478	0, 47058824	0, 46153846	-0, 42105263
0, 17659084	0, 12727273	0, 46774194	0, 11594203	0, 4109589	-0, 16363636	0, 10126582
0, 04271962	-0, 10240427	0, 21649485	0, 1442637	-0, 12396313	0, 04725125	0, 07467533
0, 00896613	-0, 31818182	-0, 17142857	0, 12195122	0, 11111111	0, 31034483	0
0, 08712121	-0, 5	1	0, 36363636	0, 25	-0, 5	-0, 09090909
0, 04082689	0, 27586207	-0, 17241379	-0, 29166667	0	0, 29032258	0, 14285714
-0, 00203134	-0, 1372549	0, 3877551	-0, 07	-0, 00990099	-0, 20183486	0, 01904762
-0, 0970078	-0, 390625	0, 31707317	-0, 29333333	0, 24637681	-0, 12307692	-0, 33846154
0, 05555556	1	0	-1	-0, 33333333	1	-0, 33333333
0, 23864379	0, 17647059	0	-0, 35294118	1	-0, 125	0, 73333333
0,	0, 4673913	-0,	0,	0,	-0,	0,

19942828		15584416	26470588	38888889	07843137	30985916
-0, 00762483	0	-0, 33333333	-0, 46153846	-0, 06666667	0, 31578947	0, 5
0, 08396887	-0, 01144165	0, 19877049	0, 25764192	-0, 04285714	0, 08914729	0, 0125523
-0, 06827485	0, 14285714	-0, 28571429	-0, 52631579	-0, 19047619	0, 25	0, 2
-0, 08280229	0, 375	-0, 4375	0, 25	-0, 16666667	-0, 11764706	-0, 4
-0, 03177644	-0, 10178117	-0, 00324675	-0, 02283105	-0, 16193182	0, 08579882	0, 01333333
-0, 03000702	-0, 23694779	0, 11707317	-0, 02164502	-0, 28571429	0, 01145038	0, 23574145
-0, 29861111	0	0	-0, 25	-0, 66666667	-0, 25	-0, 625
0, 1223055	-0, 01769912	0, 39805825	0	0, 31623932	0, 05376344	-0, 01652893
0, 01683502	0, 27272727	0, 27272727	0	0, 18181818	-0, 44444444	-0, 18181818
0, 0150619	0, 00296736	0, 08256108	0, 06286428	-0, 01664477	-0, 06450288	0, 02312634
-0, 04166667	0, 16666667	0, 42857143	0, 625	-0, 66666667	-0, 375	-0, 42857143
-0, 33333333	1	-0, 5	-1	-0, 5	-0, 5	-0, 5
0, 0777787	0, 09090909	0, 25	-0, 5	-0, 35714286	0, 44444444	0, 53846154
0, 07189598	0, 0472973	0, 07334963	0, 14798206	0, 18779343	0, 00456621	-0, 02961276
0, 12761823	-0, 22580645	0, 58823529	0, 47058824	-0, 34615385	0, 15384615	0, 125
-0, 05555556	-0, 4	0, 4	0	-0, 4	0, 4	-0, 33333333
-0, 00410593	0, 13043478	-0, 02312139	0, 14379085	-0, 0046083	-0, 23498695	-0, 03614458
0, 21326529	0, 1372549	-0, 03225807	0, 4676259	-0, 14068441	0, 44765343	0, 4
0, 37777778	0, 66666667	1	1	0	0	-0, 4
0, 15999044	0, 16129032	0, 58	-0, 18823529	-0, 12345679	0, 43157895	0, 09876543
-0,	-0,	-0,	-0, 58	0,	-0,	-0,

22152227	26530612	36538462		30188679	13461539	28571429
0, 33333333	-0, 4	-0, 4	0, 4	1	1	0, 4

Partie V. Références bibliographiques

- [ADO 05] ADOMAVICIUS G., and TUZHILIN A., Toward the Next Generation of Recommender Systems : A Survey of the State-of-the-Art and Possible Extensions, IEEE Trans. on Knowl. and Data Eng., Vol. 17 Piscataway, NJ, USA : IEEE Educational Activities Department, pages 734—749, June 2005.
- [BAS et al. 98] BASU C., HIRSH H., COHEN W., « Recommendation as classification : using social and content-based information in recommendation », Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence, American Association for Artificial Intelligence, p. 714-720, 1998.
- [BEL 92] BELKIN N. J., CROFT W. B., Information retrieval and information filtering : Two sides of the same coin? CACM, Pages 29-38, 1992.
- [BIL 98] BILLSUS D., PAZZANI M., « Learning Collaborative Information Filters » ICML '98 : Proceedings of the Fifteenth International Conference on Machine Learning, San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., p. 46-54, 1998.
- [BIL 99] BILLSUS D., PAZZANI M., A Hybrid User Model for News Story Classification, Proceedings of the Seventh International Conference on User Modeling (UM '99), Banff, Canada, June 20-24, 1999.
- [BIS 06] BISHOP C., Pattern Recognition and Machine Learning, Springer, 2006.
- [BOT et al. 03] BOTTRAUD J.C., BISSON G., BRUANDET M.F., Apprentissage de profils pour un agent de recherche d'information, Actes de la Conférence Apprentissage (CAP 2003), Laval. P 31-46, 1-2 juillet 2003.
- [BOUG et al. 04] BOUGHANEM, M., TEBRI, H., TMAR, M. : « Apprentissage par renforcement dans un système de filtrage adaptatif ». In CORIA Pages 27-40, 2004.
- [BOUT et al. 04] BOUTILIER C., BRAFMAN R., DOMSHLAK C., HOOS H., POOLE D.. CP-nets : A tool for representing and reasoning with conditional ceteris paribus preference statements. Journal of Artificial Intelligence Research, 21, 135-191, 2004.
- [BOUY 02] BOUYSSOU D., PIRLOT M., Nontransitive decomposable conjoint measurement, Journal of Mathematical Psychology, vol 46, Pages 677-703, 2002.
- [BOUY 04a] BOUYSSOU D., PIRLOT M., 'Additive difference' models without additivity and subtractivity. Journal of Mathematical Psychology, 48(3) Pages 263--291, 2004.
- [BOUY 04b] BOUYSSOU D., PIRLOT M., Preferences for multi-attributed alternatives : Traces, dominance, and numerical representations, J. of Mathematical Psychology 48, 167_185, 2004.
- [BOUY et al. 06] BOUYSSOU D., DUBOIS D., PIRLOT M., PRADE H., Concept et méthodes pour l'aide à la décision 3 - analyse multicritère, Lavoisier, 2006.
- [BOUZ 05] BOUZEGHOUB M., KOSTADINOV D., Personnalisation de l'information : aperçu de l'état de l'art et définition d'un modèle flexible de profils", Actes de la 2ème Conférence en Recherche d'Informations et Applications (CORIA), Grenoble, France, pages 201-218, 2005.
- [BOUZ et al. 07] BOUZEGHOUB M., CALABRETTO S., DENOS N., HARRATHI R., KOSTADINOV D., NGUYEN A., PERALTA V., Accès personnalisé aux informations : approche dirigée par la qualité, Informatique des Organisations et Systèmes d'Information et de Décision (INFORSID), pages 105-120, 2007.

[BRA et al. 84] BRANS J.-P., MARESCHAL B., VINCKE Ph., PROMETHEE : a New Family of Outranking Methods in Multicriteria Analysis. In : Operational Research '84. Edited by J.-P. Brans. Amsterdam : Elsevier Science Publishers B.V. pp. 408-421, 1984.

[BRE et al. 09] BREZILLON P., BREZILLON J., POMEROL J.-Ch. : « Context-based improvement of decision making : Case of car driving », International Journal of Decision Support Systems and Technology, vol. 13, pp. 1-20, 2009.

[BREE et al. 98] BREESE J.S., HECKERMAN D., KADIE C., « Empirical Analysis of Predictive Algorithms for Collaborative Filtering, » Proc. 14th Conf. Uncertainty in Artificial Intelligence, p. 43-52, July 1998.

[BRI 98] BRIN S., PAGE, L., The Anatomy of a Large-Scale Hypertextual Web Search Engine, Proceedings of the seventh international conference on World Wide Web 7, Amsterdam, The Netherlands: Elsevier Science Publishers B. V., p. 107—117, Avril 1998.

[BUR 00] BURKE R., Knowledge-based recommender systems. In A. Kent (ed.), Encyclopedia of Library and Information Systems. Volume 69, Supplement 32. Marcel Dekker, 2000.

[BUR 02] BURKE R., "Hybrid Recommender Systems : Survey and Experiments," User Model. User Adapt. Inter. vol. 12 pp. 331-370, 2002.

[CHE 06] CHEN A.Y., McLEOD D., « Collaborative Filtering for Information Recommendation Systems ». In M. Khosrow-Pour (Ed.), Encyclopedia of E-Commerce, E-Government, and Mobile Commerce Hershey, PA : Idea Group Reference, pages 118-123, 2006.

[DAN 07] DANGAUTHIER P.-C., Fondations, méthode et applications de l'apprentissage bayésien, Thèse INPG préparée au Laboratoire d'Informatique de Grenoble et à l'INRIA Rhône-Alpes, 2007.

[DEB 60] DEBREU G., Topological methods in cardinal utility theory, dans K.J. Arrow, S. Karlin, P. Suppes (dir.), Mathematical Methods in the Social Sciences, Pages 16-26, Stanford University Press, Stanford, 1960.

[DECO 93] DECO G., EBMAYER J., Coarse Coding Resource-Allocating Network. Neural Computation, Vol. 5, N° 1. pp. 105-114, 1993.

[DEC et al. 05] DECOSTE D., GLEICH D., KASTURI, T., KEERTHI, S., MADANI, O., PARK, S., PENNOCK, M. D., PORTER, C., SANGHAI, S., SHAHNAZ, F., ZHUKOV, L., Recommender Systems Research at Yahoo! Research Labs. Proceedings of Beyond Personalization 2005 : A Workshop on the Next Stage of Recommender Systems Research IUI'05. San Diego, California, USA, pp. 91-92, 2005.

[DE JONG 75] DE JONG, K.-A. An Analysis of the Behaviour of a class of Genetic Adaptive Systems. Thèse de doctorat, University of Michigan, Ann Arbor, Michigan. 266 p, 1975.

[DON 1992] DONG C.. Développement d'outils d'aide à la décision dans des environnements multicritères, dynamiques et incertains : Application à des problèmes de planification régionale. Thèse Doct. Ing. : Université Catholique de Louvain, Louvain-la-Neuve, Belgique. 169 p, 1992.

[DUB 86] DUBOIS D. and PRADE H.. Weighted minimum and maximum operations in fuzzy set theory. Information Sciences, 39 :205-210, 1986.

[EDW 71] EDWARDS W., Social utilities, Engineering Economist, Summer Symposium Series, vol 6, p 119-129, 1971.

[FAR 99] FARGIER H., PERNY P., Qualitative Decision Models under Uncertainty without the commensurability hypothesis In : K.B. Laskey and H. Prade (eds), Proceedings of Uncertainty in Artificial Intelligence, pages 188-195. Morgan Kaufmann Publishers, 1999.

[FARQ 81] FARQUHAR P.H., Multivalent preference structures, Mathematical Social Sciences, vol 1, Pages 397-408, 1981.

[FIG et al 05] FIGUEIRA, J., GRECO, S., ERGHOTT, M. (eds.) Multiple Criteria Decision Analysis: State of the Art Surveys, Springer, Berlin, pp. 507–563, 2005.

[FIS 70] FISHBURN P.C., Utility theory for decision making, John Wiley and Sons, New-York, 1970.

[FIS 75] FISHBURN P.C., Nondecomposable additive conjoint measurement for bisymmetric structures, Journal of Mathematical Psychology, vol 12, Pages 75-89, 1975.

[FIS 91] FISHBURN P.C., Nontransitive additive conjoint measurement, Journal of Mathematical Psychology, vol 35, n°1, Pages 1-40, 1991.

[FIS 95] FISHBURN P.C., WAKKER P.P., The invention of the independence condition for preferences, Management Science, vol 41, n°7, Pages 1130-1144, 1995.

[FOD 94] FODOR J., ROUBENS M., Fuzzy Preference Modelling and Multicriteria Decision support, Kluwer Academic Publishers, 1994.

[FRA et al. 95] FRÄMLING K., GRAILLOT D., INSTALLÉ M. Apprentissage interactif des fonctions de préférences par réseaux de neurones. Résumés des communications du congrès FRANCORO'95. Mons, Belgique, p. 41, 11-14 Juin 1995.

[FRA 96] FRÄMLING K., Modélisation et apprentissage des préférences par réseaux de neurones pour l'aide à la décision multicritère, Thèse pour L'INSTITUT NATIONAL DES SCIENCES APPLIQUÉES DE LYON, 1996.

[FRAN et al. 10] FRANK E., HALL M., HOLMES G., KIRKBY R., PFAHRINGER B., WITTEN I.-A., and TRIGG L., Data Mining and Knowledge Discovery Handbook, Springer, p. 1269-1277, 2010.

[FRE 07] FREDRICKS G.-A. and NELSEN R.-B., On the relationship between spearman's rho and kendall's tau for pairs of continuous random variables. Journal of Statistical Planning and Inference, 137(7), pages 2143–2150, July 2007.

[FUH 91] FUHRKEN G., RICHTER M.K., Additive utility, Economic Theory, vol 1, Pages 83-105, 1991.

[FUR 05] FÜRNKRANZ J., HÜLLERMEIER E., « Preference Learning », Künstliche Intelligenz, vol 19, Pages 60–61, 2005.

[FUR 10] FÜRNKRANZ J., HÜLLERMEIER E., « Preference Learning », (eds.), Springer, 2010.

[GAR 05] GARDEN M., DUDEK G., Semantic feedback for hybrid recommendations in recommendz. In Proceedings of the IEEE International Conference on e-Technology, e-Commerce, and e-Service (EEE05), 2005.

[GAR 06] GARDEN M., DUDEK G., Mixed Collaborative and Content-Based Filtering with User-Contributed Semantic Features. Proceedings of the 21st AAAI National Conference on Artificial Intelligence (AAAI'06), Boston, Massachusetts, July 2006.

[GAU et al. 03] GAUCH S., CHAFFE J., PRETSCHNER A., Ontology-Based User Profiles for Search and Browsing. To appear in J. User Modeling and User-Adapted Interaction : The Journal of Personalization Research , Special Issue on User Modeling for Web and Hypermedia Information Retrieval, 2003.

[GOW 03] GOWAN (Mc) J.P , A multiple model approach to personalised information access. Master Thesis in computer science, Faculty of science, University College Dublin, February 2003.

[GRA 95a] GRABISCH M., Fuzzy integral in multicriteria decision making, Fuzzy Sets and Systems, vol 69, n°3, Pages 279-298, 1995.

[GRA 95b] GRABISCH M., On equivalence classes of fuzzy connectives : The case of fuzzy integrals, IEEE Transactions on Fuzzy Systems, vol 3, n°1, Pages 96-109, 1995.

[GRA 96] GRABISCH M., The application of fuzzy integrals in multicriteria decision making, *European Journal of Operational Research*, vol 89, n°3, Pages 445-456, 1996.

[GRA 97] GRABISCH M., Alternative representations of discrete fuzzy measures for decision making, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 5(5), pages 587_607, 1997.

[GRA 98] GRABISCH M., Fuzzy integrals as a flexible and interpretable tool of aggregation, dans *Aggregation and Fusion of Imperfect Information*, Physica-Verlag, Heidelberg, *Studies in Fuzziness and Soft Computing*, vol 12, Pages 51-72, 1998.

[GRA et al. 00] GRABISCH M., MUROFUSHI T., SUGENO M. (dir.), *Fuzzy Measures and Integrals, Theory and Applications*, Physica-Verlag, Heidelberg, *Studies in Fuzziness and Soft Computing*, vol 40, 2000.

[GRA 02] GRABISCH M., PERNY P., Agrégation multicritère. In B. Bouchon-Meunier and C. Marsala, editors, *Logique floue, Principes, Aide à la décision, IC²*, pages 82–120. Hermès, 2002.

[GRA 04] GRABISCH M., LABREUCHE C., Fuzzy measures and integrals in MCDA, in J. Figueira, S. Greco & M. Ehrgott, eds, *'Multiple Criteria Decision Analysis'*, Springer, pp. 563_608, 2004.

[GRA 06] GRABISCH M., L'utilisation de l'intégrale de Choquet en aide multicritère à la décision, *Newsletter of the European Working Group "Multicriteria Aid for Decisions"*, Vol 3 No 14, pages 5–10, 2006.

[GRA et al. 08] GRABISCH M., KOJADINOVIC I., MEYER P., A review of methods for capacity identification in Choquet integral based multi-attribute utility theory : Applications of the Kappalab R package. *European Journal of Operational Research* 186(2) : pages 766-785, 2008.

[GRE 85] GREFENSTETTE J., Optimization of Control Parameters for Genetic Algorithms. *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 16, N° 1. pp. 122-128, 1985.

[GRE 87] GREFENSTETTE J., Multilevel Credit Assignment in a Genetic Learning System. In : *Proceedings of the Second International Conference on Genetic Algorithms and Their Applications*. Cambridge, MA : Lawrence Erlbaum Publishers. pp. 202-209, 1987.

[HAM et al. 07] HAMDADOU D., LABED K., BENYETTOU A. "Un Système Interactif Multicritère d'Aide à la Décision en Aménagement du Territoire : Approche du Tri, Intégrale de Choquet et SIG" SETIT 2007 4th International Conference : Sciences of Electronic, Technologies of Information, and Telecommunications , TUNISIA, March 25-29, 2007.

[HEB 49] HEBB D., *The Organization of Behaviour*. New-York : Wiley, 1949.

[HER et al. 04] HERLOCKER J., KONSTAN J., TERVEEN L., REIDL J., Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems* 22, pages 5–53, 2004.

[HOA et al. 99] HOASHI K., MATSUMOTO K., INOUE N., HASHIMOTO K., Experiments on the TREC-8 filtering track. *Proceedings of TREC-8*, 1999.

[HOLL 75] HOLLAND, J.H. *Adaptation in Natural and Artificial Systems*. Ann Arbor, Michigan : University of Michigan Press. 183 p, 1975.

[HOL et al. 94] HOLMES G., DONKIN A., WITTEN I. H., "Weka : a Machine Learning Workbench." *Intelligent Information Systems*, 1994. *Proceedings of the 1994 Second Australian and New Zealand Conference on (August 1994)* , p. 357-361., 1994.

[HUA 08]. HUANG S. L. Comparision of utility-based recommendation methods. In *The Pacific Asia Conference on Information Systems*, paper 21, 2008.

[HUL 97] HULL D. A., The TREC-6 filtering track : Description and analysis. *Proceedings of TREC-6*, 1997.

- [HUL 98] HULL D. A., The TREC-7 filtering track : Description and analysis. Proceedings of TREC-7, 1998.
- [HUN 99] HUHNS M. N., STEPHENS L. M., Personal Ontologies, Internet Computing, Vol. 3, No. 5, pp.85-87, Sept./Oct. 1999.
- [HWA 09] HWANG S and LEE M.-W., A Uncertainty Perspective on Qualitative Preference, Uncertainty Management in Information Systems, ser. Dagstuhl Seminar Proceedings, 2009.
- [INS 09] Insites consulting, Les achats en ligne ont changé radicalement en 2009, insites.eu/media/4673/communiquedepresseinsites_mc%20dc_fr.pdf, 2009.
- [JAC 81] JACQUET-LAGRÈZE E.. Systèmes de décision et acteurs multiples. Contribution à une théorie de l'action pour les sciences des organisations. Thèse d'état : Université Paris-Dauphine. 606 p, 1981.
- [JOA 02] JOACHIMS T., Optimizing search engines using clickthrough data. In ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), pp. 133–142, 2002.
- [JOK 91] JOKINEN P., Continuously Learning Nonlinear Networks with Dynamic Capacity Allocation. Thèse Doct. : Tampere University of Technology. 100 p, 1991.
- [KEE 76] KEENEY R. L., RAIFFA H., Decisions with Multiple Objects. John Wiley and Sons, New York, 1976.
- [KEE 93] KEENEY R. L., RAIFFA H., Decisions with Multiple Objectives : Preferences and Value Tradeoffs, Cambridge University Press, Cambridge, 1993. Version originale : John Wiley and Sons, New York, 1976.
- [KEN 38] KENDALL M. G., A new measure of rank correlation. Biometrika, 30(1–2), pages 81–93, 1938.
- [KEN 45] KENDALL M. G.. The treatment of ties in ranking problems. Biometrika, 33(3), pages 239–251, 1945.
- [KOD 90] KODRATOFF Y., MICHALSKI R.. Machine Learning Volume III. San Mateo California : Morgan Kaufmann Publishers, Inc.. 825 p, 1990.
- [KOJ 06] KOJADINOVIC I., Contributions à l'interprétation de mesures non additives et à l'identification de modèles décisionnels fondés sur l'intégrale de Choquet, Mémoire HDR pour l'Université de Nantes, 21 Novembre 2006.
- [KOJ 07] KOJADINOVIC I.. Minimum variance capacity identification. European Journal of Operational Research, 177(1), pages 498–514, 2007.
- [KOL 87] KOLODNER, J.-L. Extending Problem Solver Capabilities Through Case-Based Inference. Proceedings of the Fourth International Workshop on Machine Learning, University of California, Irvine, June 22-27, 1987. Los Altos (CA) : Morgan Kaufmann Publishers, Inc., pp. 167-178, 1987.
- [KRA et al. 71] KRANTZ D., LUCE R., SUPPES, P. & TVERSKY, A., Foundations of measurement, volume 1 : Additive and polynomial representations, Academic Press, 1971.
- [KWO et al. 00] KWOK K., GRUNFELD L., CHAN M., DINSTL N., Trec-9 cross language,web and question-answering track experiments using pirs. In Proceedings of the 10th Text REtrieval Conference (TREC-9). Gaithersburg, Maryland, 2000.
- [LEM 11] LEMOUZY S., Systèmes interactifs auto-adaptatifs par systèmes multi-agents auto-organisateurs : application à la personnalisation de l'accès à l'information, Thèse pour l'Université Paul Sabatier, 2011.
- [LIN 89] LIN L. I., A concordance correlation coefficient to evaluate reproducibility, Biometrics, 45, pages 255-268, 1989.

[LIND et al. 03] LINDEN G., SMITH B., and YORK J.. Amazon.com Recommendations : Item-to-Item Collaborative Filtering. IEEE Internet Computing, volume 7, pages 76-80, Jan.-Feb. 2003.

[LIU 09] LIU T.-Y., Learning to Rank for Information Retrieval, Foundations and Trends in Information Retrieval : Vol. 3 : No 3, pp. 225–331, 2009.

[LUC 64] LUCE R.D., TUKEY J.W., , Simultaneous conjoint measurement : A new type of fundamental, measurement, Journal of Mathematical Psychology, vol 1, Pages 1-27, 1964.

[MAR 00a] MARISHAL J.L., An axiomatic approach of the discrete Choquet integral as a tool to aggregate interacting criteria, IEEE Transactions on Fuzzy Systems 8(6), pages 800-807, 2000.

[MAR 00b] MARISHAL J.L., ROUBENS M., 'Determination of weights of interacting criteria from a reference set', European Journal of Operational Research 124, 641-650, 2000.

[MART et al. 12a] MARTIN A., ZARATE P., CAMILLERI G., Gestion et évolution de profils Multicritères de décideur par apprentissage pour l'aide à la Décision, 30ème congrès Inforsid, Inforsid 2012 , Pages 223 à 238, 2012.

[MART et al. 12b] MARTIN A., ZARATE P., CAMILLERI G., Decision Support Systems : Management and Evolution of Multi-Criteria Decision Makers' Profile, DSS2012, 16th IFIP WG8.3 International Conference on Decision Support Systems, Fusing DSS into the Fabric of the Context, Supplemental proceedings ISSN 1647-7103, 2012.

[MAY et al. 94] MAYSTRE L.-Y., PICTET J., SIMOS J., Méthodes multicritères ELECTRE. Lausanne : Presses polytechniques et universitaires romandes. 323 p, 1994.

[MEY et al. 06] MEYER P., GRABISCH M., KOJADINOVIC I., Kappalab : an R package for Choquet integral based MAUT, 63rd meeting of the EURO Working Group Multicriteria Aid for Decisions (MCDA), Porto, Portugal, 30-31 March 2006.

[MIL et al. 03] MILLER B.-N., ALBERT I., LAM S.-K., KONSTAN J.-A., RIEDL J., MovieLens Unplugged : Experiences with an Occasionally Connected Recommender System. In Proceedings of ACM 2003 International Conference on Intelligent User Interfaces (IUI'03) (Accepted Poster), pages 263-266, January 2003.

[MOO 00] MOONEY R. J., ROY L., Content-based book recommending using learning for text categorization. In Proceedings of the Fifth ACM Conference on Digital Libraries, pages 195-204. ACM Press, 2000.

[MOR 89] MORI T., MUROFUSHI T., An Analysis of Evaluation Model Using Fuzzy Measure and the Choquet Integral. In 5th Fuzzy System Symposium, Kobe, Japan, pages 207-212, 1989.

[MUR 93] MUROFUSHI T., SONEDA S., Techniques for reading fuzzy measures (III) : interaction index. In 9th Fuzzy System Symposium, pages 693–696, Sapporo, Japan, In Japanese, May 1993.

[NAK 90] NAKAMURA Y., Bilinear utility and a threshold structure for nontransitive preferences, mathematical Social Sciences, vol 19, Pages 1-21, 1990.

[NEU 44] NEUMANN J. von., MORGENSTERN, O. Theory of Games and Economic Behavior. Princeton, NJ : Princeton University Press, 1944.

[PAE 78] PAELINCK, J., Qualiflex, a flexible multiple criteria method. Economic letters, N° 3. pp. 193-197, 1978.

[PAZ 97] PAZZANI M., BILLSUS D., « Learning and Revising User Profiles : The Identification of Interesting Web Sites », Machine Learning, vol. 27, Pages 313-331, 1997.

[PLA 91] PLATT J., A Resource-Allocating Network for Function Interpolation. Neural Computation, Vol. 3. pp. 213-225, 1991.

- [POM et al. 93] POMEROL, J.-CH., BARBA-ROMERO S., Choix multicritère dans l'entreprise. Paris : Hermes. 391 p, 1993.
- [QUE et al. 07] QUEIROZ S., GONZALES C. et PERNY P. "Décision collective avec des réseaux GAI". In Congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision FRANCORO V / ROADEF 2007, pp. 217--227, Presses Universitaires de Grenoble, Grenoble, France, 2007.
- [QUI 86] QUINLAN J.R., Induction of decision trees. Machine Learning, N° 1. pp. 81-106, 1986.
- [QUI 87a] QUINLAN J.R., Decision Trees as Probabilistic Classifiers. Proceedings of 4th International Workshop on Machine Learning. Los Altos (CA) : Morgan Kaufmann Publishers. pp. 31-37, 1987.
- [QUI 87b] QUINLAN J.R., Simplifying Decision Trees. International Journal of Man-Machine Studies, N° 27. pp. 221-234, 1987.
- [QUI 90] QUINLAN J.R., Probabilistic Decision Trees. In : Machine Learning : An Artificial Intelligence Approach. Edited by Y. Kodratoff. San Mateo (CA) : Morgan Kaufmann Publishers. pp. 140-142, 1990.
- [RAI 68] RAIFFA H, Decision Analysis. Reading, Massachusetts: Addison-Wesley, 328 p., 1968.
- [RAI 69] RAIFFA H., Preferences for multi-attributed alternatives, Rapport technique n°RM-58-68-DOT/RC, The Rand Corporation, Santa Monica, Californie, 1969.
- [RAI 73] RAIFFA H, Analyse de la décision. Paris : Dunod, 328 p., 1973.
- [RAM 03] RAMOS J.. Using tf-idf to determine word relevance in document queries. In First International Conference on Machine Learning, New Brunswick :NJ, USA, Rutgers University, 2003.
- [REI 83] REICH E., "Users are individuals :- individualizing user models," Int. J. Man-Mach. Stud. vol. 18 pp. 199-214, 1983.
- [RES et al. 94] RESNICK P., IACOVOU N., SUCHAK, M., BERGSTROM, P., RIEDL, J. (1994). Grouplens : An open architecture for collaborative filtering of net- news. Proceeding CSCW '94 Proceedings of the 1994 ACM conference on Computer supported cooperative work. Pages 175-186, 1994.
- [ROB 99] ROBERTSON S.E., WALKER S., Okapi-Keenbow at TREC-8, Proceedings of TREC-8, 1999.
- [ROB 00] ROBERTSON S. E., HULL D., The TREC-9 filtering track final report. Proceeding of TREC-9, 2000.
- [ROC 71] ROCCHIO J.J , Relevance feedback in information retrieval, In the SMART retrieval system experiments in automatic document processing, Prentice Hall Inc, p. 313-323, 1971.
- [ROS 58] ROSENBLATT F., The perceptron : A probabilistic model for information storage and organization in the brain. Psychological Review, Vol. 65. pp. 386-408, 1958.
- [ROY 68] ROY, B. Classement et choix en présence de points de vue multiples, la méthode ELECTRE. Revue informatique et recherche opérationnelle, Vol. 2, N° 8. pp. 57-75, 1968.
- [ROY 71] ROY, B., BERTIER, P. La méthode ELECTRE II : une méthode de classement en présence de critères multiples. Paris : SEMA (Metra International), Direction scientifique, Note de travail N° 142. 25 p, 1971.
- [ROY 73] ROY, B., BERTIER, P. La méthode ELECTRE II : une application au media-planning. In : Operational Research. Edited by ROSS, M. Amsterdam : North-Holland Publishing Company. pp. 291-302, 1973.
- [ROY 78] ROY, B. ELECTRE III : un algorithme de classements fondé sur une représentation floue de préférences en présence de critères multiples. Cahiers du CERO, Vol. 20, N° 1, . pp. 3-24, 1978.

- [ROY 82a] ROY, B., HUGONNARD, J.-C. Classement des prolongements de lignes de métro en banlieue parisienne (présentation d'une méthode multicritère originale). Cahiers du CERO, Vol. 24, N° 2-3-4. pp. 153-171, 1982.
- [ROY 82b] ROY, B., HUGONNARD, J.-C. Le plan d'extension du métro en banlieue parisienne, un cas type de l'analyse multicritère. Les Cahiers scientifiques de la revue Transports, N° 6. pp. 77-108, 1982.
- [ROY 85] ROY B., Méthodologie multicritère d'aide à la décision, Economica, Paris, 1985. Traduction anglaise : Multicriteria Methodology for Decision Aiding, Kluwer Academic Publishers, Dordrecht, 1996.
- [ROY. 87] ROY B., VINCKE Ph., Pseudo-orders : definition, properties and numerical representation. Mathematical Social Sciences 14, pages 263-274, 1987.
- [ROY 91] ROY, B., BOUYSSOU, D. Aide à la décision fondée sur une PAMC de type ELECTRE. Paris : Université Paris-Dauphine, Document du LAMSADE, N° 69. 118 p, 1991.
- [ROY 93] ROY, B., BOUYSSOU, D. Aide à la décision : méthodes et cas. Paris : Economica, coll. « Gestion ». 695 pages, 1993.
- [ROY et al. 85] ROY, B., SKALKA, J.M. ELECTRE IS : aspects méthodologiques et guide d'utilisation. Paris : Université Paris-Dauphine, Document du LAMSADE, N° 30. 125 p, 1985.
- [RUM 86] RUMELHART D, McCLELLAND J., Parallel Distributed Processing Vol. 1. Cambridge, Massachusetts : MIT Press. 547 p, 1986.
- [SAA 84] SAATY, T. Décider face à la complexité. Paris : Entreprise moderne d'édition. 231 p, 1984.
- [SAL et al. 75] SALTON G., WONG A., YANG C.-S., A Vector Space Model for Automatic Indexing, Communications of the ACM, Vol. 18, Nr. 11 New York, NY, USA : ACM, p. 613—620, November 1975.
- [SCH et al. 07] SCHAFER J-Ben, FRANKOWSKI D, HERLOCKER J., SHILAD S., Collaborative Filtering Recommender Systems, The Adaptive Web : Methods and Strategies of Web Personalization, pages 291-324, 2007.
- [SCHA 85] SCHÄRLIG A., Décider sur plusieurs critères. Panorama de l'aide à la décision multicritère. Lausanne : Presses polytechniques et universitaires romandes. 304 pages, 1985.
- [SCHE et al. 02] SCHEIN A.I., POPESCU A., UNGAR L.H., PENNOCK D.M., « Methods and Metrics for Cold-Start Recommendations » Proc. 25th Ann. Int'l ACM SIGIR Conf., 2002.
- [SCHL 59] SCHLAIFER R., Probability and Statistics for Business Decisions. New York : McGraw-Hill. 732 p, 1959.
- [SCHL 69] SCHLAIFER R., Analysis of Decisions Under Uncertainty. New York : McGraw-Hill. 729 p, 1969.
- [SHA 53] SHAPLEY L.S., A value for n-person games. In H.W. Kuhn and A.W. Tucker, editors, Contributions to the Theory of Games, Vol. II, number 28 in Annals of Mathematics Studies, pages 307–317. Princeton University Press, 1953.
- [SHAN 11] SHANI G, GUNAWARDANA A, Evaluating Recommendation Systems, Recommender Systems Handbook, Springer, pages 257-297, 2011.
- [SEN 95] SENOUNE R., Acquisition de Connaissances et Apprentissage Automatique : contribution pour le développement incrémental d'un Système à Base de Connaissances pour les situations de crise - application au domaine de l'eau -. Thèse de Docteur : Institut National des Sciences Appliquées de Lyon (École Nationale Supérieure des Mines de Saint-Etienne), France, 189 p, 1995.
- [SIM 55] SIMON, H, A behavioural model of rational choice. Quarterly Journal of Economics , Vol. 69. pp 99-118, 1955.

- [SIM 83] SIMON, H., Administration et Processus de décision. Paris : Economica, coll. « Gestion ». 321 p, 1983.
- [SLO 94] SLOWINSKI R., Rough Set Analysis of Multi-Attribute Decision Problems. In : Rough Sets, Fuzzy Sets and Knowledge Discovery. Edited by Wojciech P. Ziarko. London : Springer-Verlag. pp. 136-142, 1994.
- [SOB 06] SOBECKI J., "Implementations of Web-based Recommender Systems Using Hybrid Methods", IJCSA Volume 3 Issue 3 : pages 52-64, 2006.
- [SU 92] SU L.T., Evaluation mesures for interactive information retrieval. Information Processing and Management IPM 28, 4, Pages 503-516, 1992.
- [SUG 74] SUGENO M., Theory of fuzzy integrals and its applications, Thèse de doctorat, Tokyo Institute of technology, Japon, 1974.
- [SUG 77] SUGENO M., Fuzzy measures and fuzzy integrals – A survey, dans Fuzzy Automata and Decision Processes, Pages 89-102, North-Holland, New York, 1977.
- [SUT 98] SUTTON R, BARTO A., Reinforcement Learning : An Introduction (Adaptive Computation and Machine Learning). The MIT Press, mars 1998.
- [TEB 04] TEBRI H., Formalisation et spécification d'un système de filtrage incrémental d'information. Thèse de doctorat, Université Paul Sabatier, décembre 2004.
- [TMA 02] TMAR M., BRAUN T., ROCA V., Modèle auto-adaptatif de Filtrage d'Information : Apprentissage incrémental du profil et de la fonction de décision, Thèse de l'Université Paul Sabatier de Toulouse, IRIT, 2002.
- [TVE 69] TVERSKY A., Intransitivity of preferences, Psychological Review, vol 76, Pages 31-48, 1969.
- [VIN 80] VINCKE P., Vrai, quasi, pseudo et precritères dans un ensemble fini, propriétés et algorithmes. Technical report, Cahiers du LAMSADE, N° 27, Université Paris Dauphine, Paris, 1980.
- [VIN 89] VINCKE P., L'aide multicritère à la décision. Bruxelles : Éditions de l'Université de Bruxelles. 179 pages, 1989.
- [WAK 89] WAKKER P.P., Additive representations of Preferences : A New Foundation of Decision Analysis, Kluwer Academic Publishers, Dordrecht, 1989.
- [WAN 92] WANG J., MALAKOOTI B., A feedforward neural network for multiple criteria decision making. Computer and Operations Research, Vol. 19. pp 151-167, 1992.
- [WEI 91] WEISS S., KULIKOWSKI C., Computer Systems That Learn. San Mateo : Morgan Kaufmann Publishers, Inc., 223 p., 1991.
- [WID 60] WIDROW B., HOFF M.E., Adaptive switching circuits. 1960 IRE WESCON Convention Record. New-York. pp. 69-104, 1960.
- [YAO 95] YAO Y. Y., Measuring retrieval effectiveness based on user preference of documents. J. Amer. Soc. Inf. Sys, 46(2), pages 133–145, 1995.
- [YAG 88] YAGER R.R., On ordered weighted averaging aggregation operators in multicriteria decision making. IEEE Trans. Systems, Man & Cybern., 18, pages 183–190, 1988.
- [YU 92a] YU W., Aide multicritère à la décision dans le cadre de la problématique du tri : concepts, méthodes et applications. Thèse de doctorat, UER Sciences de l'organisation, Université de Paris-Dauphine. 201 p, 1992.

[YU 92b] YU W., ELECTRE TRI. Aspects méthodologiques et manuel d'utilisation. Paris : Université Paris-Dauphine, Document du LAMSADE, N° 74. 80 p, 1992.

[ZHAI et al. 98] ZHAI C., JANSEN P., STOICA E., GROT N., EVANS D., Threshold Calibration in Clarit Adaptive Filtering. Proceedings of TREC-7, Pages 149-156, 1998.

[ZHAN 04] ZHANG Y., Using bayesian priors to combine classifiers for adaptive filtering, In Proceedings of 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 354–352. Sheffield UK, 2004.

[ZHAN et al. 02] ZHANG Y., CALLAN J., MINKA T., "Novelty and Redundancy Detection in Adaptive Filtering, " Proc. 25th Ann. Int'l ACM SIGIR Conf., pp. 81-88, 2002.